Martin Slota

Updates of Hybrid Knowledge Bases

Dissertação para obtenção do Grau de Doutor em
Engenharia Informática

Orientador : Doutor João Alexandre Carvalho Pinheiro Leite,
Universidade Nova de Lisboa

Júri:

Presidente: Doutor José Júlio Alves Alferes

Arguentes: Doutor Thomas Eiter
Doutor Andreas Herzig

Vogais: Doutor João Emílio Segurado Pavão Martins
Doutor José Júlio Alves Alferes
Doutor Eduardo Leopoldo Fermé
Doutor João Alexandre Carvalho Pinheiro Leite

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

October, 2012

Updates of Hybrid Knowledge Bases

To my parents

Acknowledgements

First and foremost, I would like to thank Carroline for being a great and understanding partner along this journey, and my parents for all their selfless care and support throughout my life.

I deeply appreciate the expertise and dedication of my supervisor, Prof. João Leite. He has always been ready to provide priceless help and advice in both scientific and everyday matters, and has given me freedom to explore and learn from my own mistakes. Thank you for teaching me so much during these years. I could not have wished for a better mentor *and* friend in the same person.

I am very grateful to the members of my Advisory Committee, Prof. José Alferes and Prof. Thomas Eiter, for their comments and suggestions throughout my studies.

The kind comments of the members of the Jury, Prof. José Alferes, Prof. Thomas Eiter, Prof. Eduardo Fermé, Prof. Andreas Herzig, Prof. João Leite and Prof. João Pavão Martins, have given me additional food for thought and helped improve the thesis, resulting in this final version of it. I highly appreciate the time and effort they invested in evaluating my work.

I thank Marco Alberti, Alfredo Gabaldon, Sofia Gomes, Ricardo Gonçalves, Martin Homola, Matthias Knorr and Terrance Swift for fruitful cooperation and joint work.

For valuable discussions about parts of this work I would also like to thank Martin Baláž, Alexander Bochman, David Pearce and Ján Šefránek.

I greatly appreciated all the seminars and discussions with the members of the KRR Research Group at Centro de Inteligência Artificial. Thank you for sharing your knowledge and experience with me. A special thanks belongs to the fellow students Sofia Gomes and João Moura for being exceptional companions during conferences.

I would also like to acknowledge the Departamento de Informática, and the Centro de Inteligência Artificial, for the working conditions provided.

Additionally, I highly appreciate the invariably kind, helpful and professional attitude of the Secretarial Staff.

My friend The Anh Han has been an excellent flatmate and travelling companion, and I would like to thank him for all the great food and interesting conversations. I also took great pleasure in the joint activities with the fellow students Rasha Ahmaed, Sofia Gomes, The Anh Han, Matthias Knorr, João Moura, Mishiko Rukhaia and Ari Saptawijaya.

Carlos Rebelo was certainly the most memorable landlord in my life and I am indebted to him for all advice and assistance he has provided.

The very warmest of thanks goes to all the wonderful Ultimate Frisbee people in Portugal, and especially to David, Margot, Mimi, Nelson, Patrick, Puto, Ricardo, Sofia and

Tiago. The trainings, pick-up games and tournaments were a joyful and very important part of my life during this challenging period.

This work was supported by FCT scholarship (SFRH/BD/38214/2007) and conceived within the FCT funded project ERRO (PTDC/EIA-CCO/121823/2010).

Abstract

Increasingly many real world applications need to intelligently access, combine and reason with large amounts of dynamically changing and highly interconnected information. The vision of the Semantic Web is to foster such advanced new applications by imbuing Web content with machine-processable metadata. Throughout the last decade, two distinct knowledge representation paradigms have been standardised to capture such metadata: *ontology languages* based on Classical Logic and *reasoning rules* based on Logic Programming. Both offer important features for knowledge representation and the interest in their integration has recently resulted in frameworks for *hybrid knowledge bases* that consist of an ontology and a rule component. However, despite the abundance of work on dynamics of ontologies and rules, when taken separately, evolution of hybrid knowledge bases has not been addressed. The goal of this thesis is to tackle this problem, focusing on *updates* of hybrid knowledge bases. A principled, formal understanding of updates is particularly interesting as it paves the way towards automated support for dealing with the vast amount of constantly changing information on the Web.

In the first part of the thesis we develop two update semantics for hybrid knowledge bases that fit the needs of particular use cases of hybrid knowledge and provide the expected results when used in specific application domains. The first semantics uses a given ontology update operator to update the ontology component of a hybrid knowledge base *in the presence of static rules*. Inspired by a realistic application, the second semantics offers a way to *modularly combine* an ontology update operator with a rule update semantics. It can be used for performing updates of hybrid knowledge bases consisting of ontology and rule layers that share information through a rule-based interface. Both of these developments constitute solutions to the problem of hybrid updates for restricted classes of hybrid knowledge bases.

Subsequently, we seek to provide a general solution that could handle updates of arbitrary hybrid knowledge bases. After pinpointing serious difficulties due to the semantic incompatibilities between existing approaches to ontology and rule updates, in the second part of the thesis we look for ways to bring them closer together by developing *semantic characterisations of rule updates*. We show that the classical, semantic, approach to updates, even when applied to a very expressive semantic characterisation of logic programs, leads to the violation of essential properties of existing rule update semantics. This leads us to the development of richer semantic characterisations of logic programs. The main result of this line of work is the introduction of a generic method for specifying update operators that can capture both classical updates as well as the historically

first rule update semantics. This constitutes the first common ground for these different update paradigms and enables us to closely examine the semantic properties of rule updates.

Together, the developments introduced in this thesis make it possible, for the first time, to devise automated tools for supporting the dynamics of hybrid knowledge. At the same time they provide a unifying perspective of both ontology and rule updates, seemingly irreconcilable, deepening our understanding of their foundations, taken separately and in combination.

Keywords: Semantic Web, Knowledge Representation, Belief Change, Ontologies, Logic Programs, Hybrid Knowledge Bases, Updates

Resumo

Existem cada vez mais aplicações no mundo real que têm necessidade de aceder, combinar e raciocinar de forma inteligente com largas quantidades de informação interligada e em mutação. A Web-Semântica pretende fomentar essas novas aplicações, dotando o conteúdo da Web com meta-dados processáveis por máquinas. Ao longo da última década, dois paradigmas distintos de representação de conhecimento foram padronizados para lidar com esses meta-dados: *linguagens de ontologias* baseadas em Lógica Clássica e *regras* baseadas em Programação em Lógica. Ambos possuem características importantes para a representação de conhecimento, e o interesse na sua integração resultou, recentemente, em *paradigmas híbridos para representação de conhecimento*, contendo componentes de ontologia e regras. No entanto, apesar da abundância de trabalho sobre a dinâmica de ontologias e regras quando tomados separadamente, o problema da evolução de bases de conhecimento híbridas não foi até hoje abordado. O objectivo desta tese é o de lidar com este problema, focando na *actualização* de bases de conhecimento híbridos. A compreensão formal e baseada em princípios deste problema abrirá caminho para a automatização do suporte para lidar com a grande quantidade de informações em constante mudança na Web.

Na primeira parte da tese desenvolvemos duas semânticas de actualização para bases de conhecimento híbridas, que atendem às necessidades de casos de uso específicos e fornecem os resultados esperados quando utilizadas em domínios de aplicação específicos. A primeira semântica usa um operador de actualização de ontologias existente para actualizar a componente de ontologia de uma base de conhecimento híbrida, *na presença de regras estáticas*. Inspirado por uma aplicação realista, a segunda semântica oferece uma *forma modular de combinar* um operador de actualização de ontologias com uma semântica de actualização de regras, podendo ser usada para realizar actualizações de bases de conhecimento híbridas compostas por camadas de ontologias e regras que compartilham informações através de interfaces baseadas em regras. Ambos os desenvolvimentos constituem soluções para o problema das actualizações para classes restritas de bases de conhecimento híbridas.

Posteriormente, procuramos fornecer uma solução geral que pudesse lidar com actualizações de bases de conhecimento híbridas arbitrárias. Depois de identificar um conjunto de sérias dificuldades devido à incompatibilidade semântica entre as abordagens existentes para a actualização de ontologias e de regras, na segunda parte da tese procuramos formas de as aproximar, através do desenvolvimento de *caracterizações semânticas* *de actualizações de regras*. Mostramos que a abordagem clássica, semântica, de actualizações, mesmo quando aplicada a uma caracterização semântica de programas em lógica muito expressiva, resulta na violação de propriedades essenciais das semânticas de actualização de regras existentes. Isso leva-nos ao desenvolvimento de caracterizações semânticas de programas em lógica mais ricas. O resultado principal desta linha de trabalho é a introdução de um método genérico para especificar operadores de actualização que pode capturar tanto as actualizações clássicas bem como a primeira semântica de actualização de regras. Este constitui o primeiro terreno comum para estes paradigmas de actualização distintos, permitindo-nos examinar de perto as propriedades semânticas das actualizações de regras.

Em conjunto, os desenvolvimentos introduzidos nesta tese tornam possível, pela primeira vez, o desenvolvimento de ferramentas automáticas de suporte à dinâmica de conhecimento híbrido, fornecendo igualmente uma perspectiva unificadora das duas abordagens de actualização de conhecimento, aparentemente irreconciliáveis, com o consequente aprofundar da nossa compreensão sobre os fundamentos de actualizações de ontologias e regras, separadamente e em conjunto.

Palavras-chave: Web-Semântica, Representação de Conhecimento, Alteração de Crenças, Ontologias, Programas em Lógica, Bases de Conhecimento Híbridas, Actualizações

Contents

List of Figures

List of Tables

Part I

Ontologies, Rules and Their Updates

1 Introduction

> The greatest challenge to any thinker is stating the problem in a way that will allow a solution.
>
> Bertrand Russell
> *British author, mathematician & philosopher*

In this thesis we address the problem of updating hybrid knowledge bases which consist of both Description Logic axioms as well as Logic Programming rules. We first approach the problem from two different perspectives, both with a practical appeal, and then make theoretical contributions towards finding a general solution.

Many human endeavours and achievements are rooted in our ability to create abstractions of the environment that surrounds us. We use them on a daily basis as substitutes for the real world in order to focus on aspects that are relevant to our situation, ignoring extraneous details. Deliberation about such abstractions enables us to identify patterns where initially we could not see any, and use them to make predictions and decide upon the best course of action.

Knowledge Representation and Reasoning (KRR) (McCarthy and Hayes, 1969; Minsky, 1975; Hayes, 1979; Levesque, 1984; Davis et al., 1993) is a field of research concerned with developing means of formulating abstractions as well as systematic methods for drawing conclusions from them. Inspired by work in a wide range of research areas, such as philosophy, mathematics, computer science, psychology, biology or economics, it forms one of the major subfields of Artificial Intelligence – it enables machines to process symbolic descriptions of the world and act based on them accordingly.

During the last decade, results developed in KRR have been adopted by pioneers working to bring about the vision of the Semantic Web (Berners-Lee et al., 2001). Led by the World Wide Web Consortium (W3C),[1] an international community that develops standards to ensure the long-term growth of the Web, this movement was initiated to foster advanced new applications that need to acquire, combine and reason with information on the Web.

The World Wide Web, even in its current form, is a very powerful medium that contains vast amounts of interconnected and dynamically changing information. Existing search engines greatly facilitate the process of finding relevant content to millions of people around the world. Nevertheless, the functionality and accuracy of search engines is severely constrained by the long-standing limitations of computer algorithms for processing the nuances and ambiguities of natural language, and extracting information from audio-visual content. The principal idea of the Semantic Web is to adopt a more pragmatic approach: imbue Web content with machine-processable metadata.

Such metadata requires a representation language with sufficient expressivity, an unambiguous interpretation accessible to both men and machines, and support for reasoning procedures that are guaranteed to terminate. These necessities naturally point towards knowledge representation languages based on formal logic, initially divided in two seemingly incompatible groups: Description Logics and Logic Programs. Their reconciliation soon became one of the major challenges in Semantic Web research.

1.1 Description Logics and Logic Programs

Description Logics (DLs) (Baader et al., 2007) form the first group of logic-based representation languages adopted for Semantic Web standards. Usually, they are *fragments of first-order logic* with *decidable reasoning tasks*, and offer an expressive representation language coupled with the standard first-order semantics and sound and complete reasoning procedures. They are powerful enough to capture and extend the capabilities of existing modelling languages used in software engineering. Furthermore, they form the foundation of the W3C standard for specifying knowledge about data on the Web: the Web Ontology Language (OWL).[2] This renders Description Logics, hence Classical Logic, the standard for knowledge representation and sharing.

One of the important features of DLs is that they do not adopt the *Closed World Assumption* (CWA), as done for instance in relational databases, so they do not assume that the represented knowledge is *complete*. This is usually referred to as the *Open World Assumption* (OWA), and it essentially means that a proposition is considered false only if the knowledge base is inconsistent with it, and the possibility of existence of objects not explicitly mentioned in the knowledge base is taken into account. It is in accord with the open and distributed nature of the Web where one can hardly assume to have complete knowledge about all relevant entities.

Syntactically, a Description Logic knowledge base, frequently referred to as an *ontology* (Gruber, 1993), uses three types of basic symbols: *individuals*, representing objects, *concepts*, representing groups of objects, and *roles*, representing binary relations between objects. Typically, an ontology is composed of two distinguishable parts: a *TBox* with descriptions of concepts and roles, and an *ABox* with assertions about individuals. We illustrate these notions in the following example:

Example 1.1 (Electronic Market – Ontology)**.** Imagine an open electronic market where users can exchange resources. For instance, a user can pay for an educational course

[1] `http://www.w3.org/`
[2] `http://www.w3.org/TR/owl-overview/`

or, say, exchange computational resources for communication services. Semantic Web technologies can help process information about users, goods and services from multiple sources and mediate exchanges by implementing appropriate protocols and policies.

In our example, the internal ontology specifies that *users* can offer *services* and a *course* is one kind of service. It also contains rather obvious facts, such as that no user is a service and no service is a user. These terminological definitions can be encoded in a DL TBox using the following assertions:

$$\mathsf{Course} \sqsubseteq \mathsf{Service} \qquad \exists\mathsf{Offers} \sqsubseteq \mathsf{User}$$
$$\mathsf{User} \sqcap \mathsf{Service} \sqsubseteq \bot \qquad \exists\mathsf{Offers}^- \sqsubseteq \mathsf{Service}$$

From the formal viewpoint, these axioms describe relationships between three concepts (Course, Service and User) and one role (Offers). The ones on the left-hand side specify that a course is a special kind of service and that no individual can be both a user and a service. Those on the right-hand side indicate that the domain of the role Offers is User and its range is Service. We can equivalently translate this TBox into standard first-order logic as follows:

$$\forall \mathbf{x} : \mathsf{Course}(\mathbf{x}) \supset \mathsf{Service}(\mathbf{x}) \qquad \forall \mathbf{x} : (\exists \mathbf{y} : \mathsf{Offers}(\mathbf{x}, \mathbf{y})) \supset \mathsf{User}(\mathbf{x})$$
$$\forall \mathbf{x} : \mathsf{User}(\mathbf{x}) \land \mathsf{Service}(\mathbf{x}) \supset \bot \qquad \forall \mathbf{x} : (\exists \mathbf{y} : \mathsf{Offers}(\mathbf{y}, \mathbf{x})) \supset \mathsf{Service}(\mathbf{x})$$

The ABox in our example contains assertions about particular users and services, possibly imported from foreign ontologies:

$$\mathsf{User}(\mathit{adolf}) \qquad \mathsf{Course}(\mathit{meditation}) \qquad \mathsf{Course}(\mathit{singing})$$
$$\mathsf{Offers}(\mathit{adolf}, \mathit{meditation}) \qquad \mathsf{LanguageOf}(\mathit{meditation}, \mathit{english})$$

Inference procedures can now be invoked to infer information that is implicitly contained in the ontology. For example, $\mathsf{Service}(\mathit{meditation})$ is true because every course is a service, and $\mathsf{Service}(\mathit{adolf})$ is false because adolf is a user and no individual can be both a user and a service.

A user looking for available courses taught in English can now query this ontology. At this point, the desirability of using the Open World Assumption to reason with represented knowledge becomes noticeable. If the reasoning algorithm were to use the Closed World Assumption, as is usually the case in relational databases, it would conclude that the proposition $\mathsf{LanguageOf}(\mathit{singing}, \mathit{english})$ is false. But the absence of information about the language of the singing course does not imply that it is not taught in English. When using the Open World Assumption, the reasoning procedure will not be able to conclude that $\mathsf{LanguageOf}(\mathit{singing}, \mathit{english})$ is true nor that it is false and, possibly, include it among the courses suggested to the enquiring user.

Actually, DLs constitute a whole family of knowledge representation formalisms, each with different constraints regarding the complex concepts and roles one can form from atomic ones and with different types of TBox and ABox axioms. Among them is the prototypical DL called the *Attributive Concept Language with Complements* ($\mathcal{ALC}$). Many other DLs are based on $\mathcal{ALC}$ and their names correspond to the features they add to it. For instance, the DL $\mathcal{ALCI}$ adds inverse roles ($\mathcal{I}$) to $\mathcal{ALC}$. The above example uses only constructs allowed in $\mathcal{ALCI}$. Further details on DLs can be found in (Baader et al., 2007).

The second logic-based knowledge representation paradigm adopted for Semantic Web standards is Logic Programming. Syntactically it is based on rules that can naturally express many kinds of information and are familiar to software developers. It allows for declarative specifications that closely resemble the problem itself, and features formal, declarative and well-understood semantics, the *stable models semantics* (Gelfond and Lifschitz, 1988) and its tractable approximation, the three-valued *well-founded semantics* (Gelder et al., 1991), being the most prominent and widely accepted.

These semantics adopt the *Closed World Assumption* (CWA), i.e. knowledge is considered *complete by assumption*. Consequently, a proposition is false whenever it is not entailed to be true. This type of negation is usually dubbed *default negation* to distinguish it from negation in Classical Logic. And unlike Classical Logic, which is monotonic, default negation leads to non-monotonicity: additional knowledge may refute previous conclusions. One can thus reason with incomplete information and *stable models* offer an intuitive semantics for reasoning about several possible consistent worlds.

Efficient solvers for the stable models semantics (Syrjänen and Niemelä, 2001; Leone et al., 2006; Gebser et al., 2007) and the well-founded semantics (Rao et al., 1997) have made it possible to use Logic Programs in real applications, such as decision support for a space shuttle (Nogueira et al., 2001), automated product configuration (Soininen et al., 2001; Tiihonen et al., 2003), heterogeneous data integration (Leone et al., 2005) and inferring phylogenetic trees (Brooks et al., 2007), but also reasoning about action (Lifschitz, 1999b), planning (Subrahmanian and Zaniolo, 1995; Lifschitz, 1999a), diagnosis (Eiter et al., 1999), resource allocation (Leite et al., 2009), etc. It is also widely acknowledged that Logic Programming offers a natural way of expressing norms and laws (Sergot et al., 1986; Kowalski, 1992, 1995; Prakken and Sartor, 2002) as well as policies (Chomicki et al., 2000; Son and Lobo, 2001).

It has soon been realised that rules are fundamental to overcome the limitations found in OWL (Donini et al., 1998; Grosof et al., 2003; Motik et al., 2006). Among the most important is their ability to use default negation to locally assume that parts of represented knowledge are complete. This allows for reasoning based on the absence of information and provides a natural mechanism for expressing exceptions and defaults, as illustrated in the following example:

Example 1.2 (Electronic Market – Rules and Incomplete Information)**.** Our electronic market may need to detect different types of users to tailor advertisements to their liking. For instance, consumers can be classified as users who do not actively offer services to other users. In order to express this, a form of the CWA is needed to query for the *absence* of offers recorded in the ontology:

$$\mathsf{HasOffer}(\mathbf{x}) \leftarrow \mathsf{User}(\mathbf{x}), \mathsf{Offers}(\mathbf{x}, \mathbf{y}).$$
$$\mathsf{Consumer}(\mathbf{x}) \leftarrow \mathsf{User}(\mathbf{x}), \sim\mathsf{HasOffer}(\mathbf{x}).$$

The first rule introduces a concept HasOffer for users with a recorded offer while the second rule classifies the remaining users as members of the concept Consumer.

Rules can also be used to encode norms that frequently involve a form of presumption of innocence and thus require the CWA. For example, a norm can specify that an offer is made public only if offered by a user that has not violated the terms of service, with one possible violation being an overdue payment:

$$\mathsf{Visible}(\mathbf{y}) \leftarrow \mathsf{Offers}(\mathbf{x}, \mathbf{y}), \sim\mathsf{Violation}(\mathbf{x}). \tag{1.1}$$
$$\mathsf{Violation}(\mathbf{x}) \leftarrow \mathsf{OverduePayment}(\mathbf{x}). \tag{1.2}$$

The first rule makes an offer visible only if no violations are currently recorded and the second rule specifies that an overdue payment is a violation.

Other types of information, such as users' preferences and default behaviour of the system in the absence of a preference, can also be naturally and declaratively represented using rules.

Another important feature of Logic Programs is the ability to express integrity constraints. This is not conceivable in DLs alone since all DL assertions may have side-effects, defeating the purpose of a constraint which is to merely *check* whether represented knowledge satisfies some condition (Reiter, 1992).

Example 1.3 (Electronic Market – Constraints)**.** In the electronic market, rules can be used to enforce that an offering user for every service must be known:

$$\begin{aligned} \mathsf{HasOfferer}(\mathbf{x}) &\leftarrow \mathsf{Service}(\mathbf{x}), \mathsf{User}(\mathbf{y}), \mathsf{Offers}(\mathbf{y}, \mathbf{x}). \\ &\leftarrow \mathsf{Service}(\mathbf{x}), \sim\mathsf{HasOfferer}(\mathbf{x}). \end{aligned}$$

The first rule introduces a concept HasOfferer with services whose offering user is known. The second rule, a constraint, forbids a situation where some service does not belong to the concept HasOfferer. These two rules can detect the inconsistency in the ontology presented in Example 1.1 which arises due to a lack of information about a user that offers the $singing$ course. Notice that a TBox axiom such as

$$\mathsf{Service} \sqcap \neg\exists\mathsf{Offers}^{-} \sqsubseteq \bot$$

does not serve the intended purpose because it simply "generates" an unnamed offering user for every service that doesn't have a named one. In particular, the violation of the intended constraint in Example 1.1 would remain undetected.

The need for rules on the Semantic Web has resulted in efforts to standardise rule languages. In 2010, the W3C standard called *Rule Interchange Format* (RIF)[3] has established as a standard XML language for expressing rules on the Web.

1.2 Hybrid Knowledge Bases

Standardisation of both ontology and rule languages, such as OWL and RIF, respectively, has fostered larger and larger numbers of ontologies and rule bases with different levels of complexity and scale. As seen in the examples above, both formalisms offer important features for knowledge representation on the Web as well as in any other complex open system. Whereas ontologies provide the logical underpinning of intelligent access and information integration, rules are widely used to represent business policies, regulations and declarative guidelines about information. What was needed was a unified knowledge framework where expressivity of both formalisms could be seamlessly combined.

Such integration however turned out to be a difficult task. The principal reason for this lies in the inherent semantic differences between the Open and Closed World Assumptions which need to be reconciled when predicates are concurrently defined both by ontology axioms and using rules. It also turned out that extending even not very expressive Description Logics with recursive Horn rules, in an unconstrained manner, results in undecidability of basic reasoning tasks (Levy and Rousset, 1998).

Four clearly distinguishable approaches to tackle these problems have been adopted throughout the years (Hitzler and Parsia, 2009; de Bruijn et al., 2010). The first group

[3] http://www.w3.org/TR/rif-overview/

extends Description Logics with *monotonic* rules and adopts an overall semantics based on first-order logic. This is the case of an early approach, $\mathcal{AL}$-log (Donini et al., 1998), as well as the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004), which is a union of the Description Logic underlying OWL and (binary) function-free Horn logic without any restrictions. Decidability can be achieved by restricting the application of rules to known individuals appearing in the knowledge base (DL-safety) (Krötzsch et al., 2008b). Another similar approach proposes Description Logic Programs (Grosof et al., 2003), a Horn fragment of OWL providing a basic form of interoperability between OWL and Logic Programs. In (Krötzsch et al., 2008a), Description Logic Rules were presented, consisting of rules that allow for DL-syntax. They are restricted to certain tree-shaped structures and generalise Description Logic Programs. The main characteristic of all these frameworks is that they stay monotonic, so they do not support vital representation features of Logic Programs such as reasoning with assumptions and naturally expressing exceptions.

The second group exercises a loose, modular integration where the ontology and rules are separate from one another and new syntactic constructs are added to enable for querying the ontology from bodies of rules. This includes dl-programs (Eiter et al., 2004) and HEX-programs (Eiter et al., 2005). The resulting integration is only partial, but semantic problems are easier to handle and it is possible to reuse existing implementations.

The third group consists of frameworks that define a customised semantics for a knowledge base composed of a DL ontology and a set of rules that do not need to be monotonic, e.g. $\mathcal{DL}$-*log* (Rosati, 1999), r-hybrid and r^+-hybrid knowledge bases (Rosati, 2005a,b), G-hybrid knowledge bases (Heymans et al., 2006) and $\mathcal{DL}$+*log* (Rosati, 2006). These can also be characterised using the non-classical Quantified Equilibrium Logic (de Bruijn et al., 2010). The predicate symbols are separated in two groups, the first interpreted under the OWA and used in both the ontology and rule parts and the second interpreted under the CWA and only used in rules. Hence the integration is much tighter than in the second group but a certain degree of separation is still present.

The last group embeds both the ontology and rules in a unifying non-monotonic formalism, such as the Autoepistemic Logic (de Bruijn et al., 2011) or the Logic of Minimal Knowledge and Negation as Failure (MKNF) (Motik and Rosati, 2010). These provide a tight and semantically neat integration of the two paradigms as predicates can be viewed simultaneously under the OWA and the CWA. Currently, the more mature of these proposals are MKNF Knowledge Bases (Motik and Rosati, 2010): similarly as with SWRL, DL-safety can be imposed to guarantee decidability of reasoning, and computational complexity of various combinations of Description Logics and classes of programs has been examined. Moreover, a tractable variant of this formalism, based on the well-founded semantics (Knorr et al., 2011), allows for a top-down querying procedure and an implementation with support for the Description Logic $\mathcal{ALCQ}$ is available (Alferes et al., 2009; Gomes et al., 2010). This is an important development because the resulting formalism is still expressive enough to reason with assumptions and naturally express exceptions, and at the same time opens up the possibility of using hybrid knowledge bases in data-intensive applications.

Overall, the work on hybrid knowledge bases has matured significantly over the years and fundamental semantic as well as computational problems were addressed successfully: The proposed languages make it possible to represent and reason with hybrid knowledge. Nevertheless, there is one important aspect of the Web that is not tackled by them: its *dynamic nature*.

Participants in an open environment need to deal with a constantly changing world: new resources become available while existing ones disappear or change in unexpected ways; likewise, preferences, norms and policies naturally undergo changes with time. Accordingly, throughout their lifetime, knowledge bases are amended, debugged, revised, get updated with fresh information, or need to be merged with other bodies of knowledge. These tasks require the ability to resolve conflicts between original knowledge and its modifications without the need to rewrite the entire knowledge base from scratch.

These topics have been intensely studied in the context of Description Logics as well as in the context of Logic Programs.

1.3 Ontology Updates

The dynamics of ontologies has been studied extensively, especially during the last decade, in an area of research called *ontology change*. It encompasses a number of strongly related though distinguishable subareas, such as ontology matching, ontology integration and merging, or ontology translation (see (Flouris et al., 2008) for an overview). The purest type of change, concerned with modifications to a single ontology, is generally referred to as *ontology evolution* (Flouris et al., 2008).

A number of notably different approaches to addressing ontology evolution have been adopted, both pragmatic and theoretical. Considerable effort has been invested in the development of ontology editors, such as Protégé (Noy et al., 2000, 2006), OilEd (Bechhofer et al., 2001) or OntoEdit (Sure et al., 2003), mostly supporting low-level ontology modifications (Stojanovic and Motik, 2002). Phases of the evolution process (Stojanovic et al., 2002, 2003; Plessers and Troyer, 2005), ontologies of basic and complex change operations (Stojanovic et al., 2002; Stuckenschmidt and Klein, 2003; Klein and Noy, 2003; Noy and Klein, 2004) and user-selected strategies for implementing a change (Stojanovic et al., 2002) have also been explored. Other topics include optimisation of the selection process of axioms for user feedback (Nikitina et al., 2011) and algorithms for repairing inconsistencies and reasoning in their presence (Haase et al., 2005; Haase and Stojanovic, 2005). However, these developments lack a firm semantic underpinning, making it hard to formally analyse their behaviour and properties, thus constraining their applicability.

To overcome these limitations, many recent approaches are based on research in the area of *belief change*, initiated by the seminal work of Alchourrón, Gärdenfors and Makinson (AGM) (Alchourrón et al., 1985) who proposed a set of desirable properties of change operators on monotonic logics, now called *AGM postulates*. Subsequently, Keller and Winslett (1985), and Katsuno and Mendelzon (1991) distinguished *update* and *revision* as two very related but ultimately different belief change operations. While revision deals with incorporating new information about a *static world* into a knowledge base that might have been *incorrect* in describing it, update takes place when a knowledge base with *correct* information needs to be brought up to date when the *modelled world changes*. The following example better illustrates this distinction:

Example 1.4 (Book and Magazine (Katsuno and Mendelzon, 1991))**.** Consider a room with two objects in it, a book and a magazine. Suppose b means that the book is on the floor, and m means that the magazine is on the floor. The current state of the world is represented by a formula ϕ stating that either the book is on the floor or the magazine is, but not both. We order a robot to put the book on the floor. The result of this action should be represented by the update of ϕ with b. After the robot puts the book on the floor, all we know is b. However, according to the AGM postulates, the result of the change must entail that m is false, despite the fact that in this scenario it is not reasonable to conclude from the presented information that the magazine is not on the floor.

Based on these observations, Katsuno and Mendelzon (1991) concluded that though the original AGM postulates are desirable when making revisions, they are not appropriate for performing updates, and formulated a different set of postulates for updates (the *KM postulates for belief update*). One of the standard operators satisfying these postulates is *Winslett's update operator* (Keller and Winslett, 1985; Winslett, 1990).

Since then, research on both belief revision and update has continued. The original AGM and KM postulates have been thoroughly examined and evaluated, and additional ones have been proposed. Along with the postulates, various constructive characterisations of classes of operators that satisfy them were identified (see (Grove, 1988; Rott, 1991; Hansson, 1996; Herzig and Rifi, 1999) and references therein).

Quite early on, *ontology revision* has been recognised as a challenging problem that requires input from diverse research areas (Foo, 1995). Though belief revision could be recast to partially address revision of a single ontological concept (Wassermann, 1998; Wassermann and Fermé, 1999), it turned out that most results could not simply be carried over to deal with revision of DL ontologies. The primary reason for this is that AGM postulates were formulated for logics with properties that most DLs do not satisfy because they only allow for limited use of negation or do not (fully) support disjunction. Consequently, the postulates first needed to be reformulated accordingly (Flouris et al., 2006b,c). Unfortunately, it was found that many interesting DLs, including the ones underlying the OWL standard, are not *AGM-compliant*: no revision operator satisfying the reformulated postulates exists (Flouris, 2006). In other words, it may happen that the desired result of a revision can only be represented in a language that is more expressive than the original DL, or it must be approximated and some of the postulates sacrificed.

This fundamental issue has been approached in a variety of ways (Qi and Yang, 2008). Some developments consider revision of *disjunctive DLs* (Qi et al., 2006a,b,c; Qi and Du, 2009), diverging from mainstream DL research. Others consider approximations of revision operators (Wang et al., 2009, 2010), or provide an algorithm for deciding logical entailment from the revised knowledge base without computing it directly (Yang et al., 2009). Some approaches instead follow work on *base revision* (Hansson, 1996), i.e. they remove entire axioms from the knowledge base without retaining their original consequences (Halaschek-Wiener and Katz, 2006; Ribeiro and Wassermann, 2007; Qi et al., 2008). Though most approaches are directed at resolving inconsistency (lack of a first-order model), the problem of resolving incoherence (unsatisfiability of concepts) has also been considered (Flouris et al., 2006a; Qi et al., 2008; Qi and Du, 2009).

The work on *ontology updates* has encountered similar issues. It was found that even in the absence of TBox assertions, unrestricted ABox updates using Winslett's operator in the DL $\mathcal{ALCQI}$ lead to undecidability of basic reasoning tasks (Baader et al., 2005b). Further work on updates in DLs at least as expressive as $\mathcal{ALC}$ has thus focused on ABox updates of atomic concepts (Liu et al., 2006; Bong, 2007; Drescher et al., 2009). Another line of work has addressed ABox updates and their approximations in dialects of DL-Lite (De Giacomo et al., 2006, 2007, 2009). These seem quite promising as they allow for a static TBox and provide polynomial algorithms for computing the updates or their approximations. An ABox update of this kind is shown in the following example.

Example 1.5 (Electronic Market – ABox Update)**.** Consider the ontology of our electronic market from Example 1.1 and suppose that the meditation course has been cancelled, expressed as the ABox update

$$\neg\mathsf{Service}(meditation) \ .$$

If all previous TBox assertions must stay valid, then $\mathsf{Course}(meditation)$ must become false, otherwise we would still be able to conclude $\mathsf{Service}(meditation)$ which is inconsistent with the update. Similarly, $\mathsf{Offers}(adolf, meditation)$ must become false due to the range restriction on role Offers.

1.4 Rule Updates

When updates started to be investigated in the context of Logic Programming, it was only natural to consider adapting the belief update postulates and operators to deal with rule updates (Alferes and Pereira, 1996). However, this led to counterintuitive results, principally because one of the fundamental principles underlying belief updates is that a knowledge base is updated on the level of its models – each model is updated separately, modifying it as little as possible, and the new collection of models characterises the updated knowledge base. However, as illustrated in the following example, rules in a logic program encode essential relationships between literals which are lost when updates are performed on a model by model basis without regard to the rules that produced the models (Leite and Pereira, 1997).

Example 1.6 (Leite and Pereira (1997))**.** Consider an agent with beliefs represented by the following program $\mathcal{P}$:

$$\mathsf{GoHome} \leftarrow {\sim}\mathsf{Money}. \tag{1.3}$$

$$\mathsf{GoRestaurant} \leftarrow \mathsf{Money}. \tag{1.4}$$

$$\mathsf{Money}. \tag{1.5}$$

The only stable model of $\mathcal{P}$ is $I = \{\, \mathsf{Money}, \mathsf{GoRestaurant} \,\}$, capturing that the agent has money by rule (1.5), so according to rule (1.4) it plans to go to a restaurant. Now consider an update $\mathcal{U}$ with the following two rules:

$${\sim}\mathsf{Money} \leftarrow \mathsf{Robbed}. \qquad\qquad \mathsf{Robbed}.$$

If we update $\mathcal{P}$ by $\mathcal{U}$ following the fundamental ideas behind belief update, the result must be characterised by the stable models of $\mathcal{P}$ after they are minimally changed to become consistent with $\mathcal{U}$. And in order to make I consistent with the rules in $\mathcal{U}$, one needs to modify the truth value of two atoms, Robbed and Money, arriving at the interpretation $J = \{\, \mathsf{Robbed}, \mathsf{GoRestaurant} \,\}$. So after the update, the agent has no money but still plans to go to a restaurant. The intended result, though, is that GoRestaurant should now be false because its only justification, Money, is no longer true. Furthermore, we expect that GoHome is now true, i.e. the rule (1.3) should be triggered because Money became false.

As a result of these observations, state-of-the-art rule update semantics, in spite of being driven by the same basic intuitions and aspirations as their belief update counterparts, are based on fundamentally different principles and methods. Many adopt the *causal rejection principle* (Leite and Pereira, 1997; Alferes et al., 2000; Eiter et al., 2002; Alferes et al., 2005; Osorio and Cuevas, 2007), which states that a rule is *rejected* only if it is directly contradicted by a more recent rule. This essentially means that inertia and minimal change, applied on the level of *literals* in belief change operators, is instead applied to *rules* and the truth values of literals follow from the set of unrejected rules. Causal rejection semantics are useful in a number of practical scenarios (Alferes et al., 2003; Saias and Quaresma, 2004; Siska, 2006; Ilic et al., 2008) and their behaviour is intuitively predictable.

Example 1.7 (Electronic Market – Rule Updates)**.** Consider the ontology and rules of our electronic market from Examples 1.1 and 1.2. Suppose that we perform the following (rule) update:

$$\mathsf{OverduePayment}(adolf).$$

We can then conclude from rules (1.1) and (1.2) that $\mathsf{Violation}(adolf)$ is true and, consequently, $\mathsf{Visible}(meditation)$ is no longer true. No conflict arises after this update, so the new rule can simply be added to the original ones.

Now consider a further update that introduces the concept of an administrator into the system, declares that $adolf$ is an administrator and makes administrators immune to norm violations:

$$\mathsf{Administrator}(adolf). \qquad\qquad \sim\mathsf{Violation}(\mathbf{x}) \leftarrow \mathsf{Administrator}(\mathbf{x}).$$

After this update, a conflict arises between the latter rule and rule (1.2): the bodies of both rules are satisfied and their heads contradict each other. This conflict can be resolved by any rule update semantics based on causal rejection, concluding that $\mathsf{Violation}(adolf)$ is false and thus reinstating the truth of $\mathsf{Visible}(meditation)$.

Alternative approaches to rule updates employ syntactic transformations and other methods, such as abduction (Sakama and Inoue, 2003), forgetting (Zhang and Foo, 2005), prioritisation (Zhang, 2006), preferences (Delgrande et al., 2007), or dependencies on default assumptions (Šefránek, 2006, 2011; Krümpelmann and Kern-Isberner, 2010; Krümpelmann, 2012).

1.5 Problem: Updates of Hybrid Knowledge Bases

Starting from the research on ontology and rule updates, our goal in this thesis is to investigate *updates of hybrid knowledge bases* which, to the best of our knowledge, have never been addressed before.

Our research is guided by the following directives:

- to find solutions that fit the needs of particular use cases of updates of hybrid knowledge bases and provide the expected results when used in specific application domains;
- to seek a general solution that can be used in arbitrary use cases of updates of hybrid knowledge bases.

In the first part of the thesis we study updates in the context of *MKNF knowledge bases* (Motik and Rosati, 2010) since they provide a general and elegant semantic framework for representing and querying hybrid knowledge. We address two interesting use cases of MKNF knowledge bases and devise update semantics for them.

First we show how the static semantics for MKNF knowledge bases can be adapted to allow for *updates of the ontology component* of a hybrid knowledge base while the rule component remains static. This encompasses practical applications of hybrid knowledge bases where the ontology contains highly dynamic information and rules represent business policies, preferences or behaviour that can be overridden by ontology updates when necessary.

In the second use case we take inspiration from a realistic application in which the hybrid knowledge base can be split into *ontology and rule layers* that share information using a rule-based interface. This makes it possible to *modularly combine* an ontology update operator with a rule update semantics, and so define an update semantics for layered hybrid knowledge bases. We use generalisations of the splitting theorems for logic programs (Lifschitz and Turner, 1994) to define this modular hybrid update semantics and show that it can perform non-trivial updates and resolve conflicts *as anticipated*.

Both of these developments constitute solutions to the problem of hybrid updates for restricted classes of hybrid knowledge bases and are fully compatible with one another. Then we turn our attention to the more general problem, and ask ourselves whether these semantics can be naturally extrapolated to arrive at a universal hybrid update semantics.

As it turns out, finding a general hybrid update semantics that produces suitable results is more difficult than we initially expected. The most pressing problems are of a *semantic nature*.

Even within the area of ontology updates, while ABox updates have been successfully addressed using belief update operators, as attention moved towards TBox updates, it has been argued that Katsuno and Mendelzon's *model-based approach* does not provide suitable results when applied to TBoxes (Zheleznyakov et al., 2010; Slota and Leite, 2010b). For illustration, consider the following example:

Example 1.8 (Electronic Market – TBox Updates)**.** Take the TBox from Example 1.1 and suppose that we want to introduce a new concept, LanguageCourse, using the TBox update

$$\mathsf{LanguageCourse} \sqsubseteq \mathsf{Course} \enspace .$$

If we perform this update on the original TBox using Winslett's update operator, we obtain a new TBox which no longer entails the original TBox axiom

$$\mathsf{Course} \sqsubseteq \mathsf{Service} \enspace .$$

This seems to be very counterintuitive since the update was only meant to introduce a new concept into the concept hierarchy and there does not appear to be any reason to modify its original structure. The expected result would be to simply add the new axiom to the original TBox.

As a consequence of these observations, Zheleznyakov et al. (2010) and Calvanese et al. (2010) use the antipole of model-based operators, dubbed *formula-based*, for performing TBox updates. Recently, Lenzerini and Savo (2011) have used similar ideas to tackle ABox updates, too. What is interesting about these developments is that the resulting operators bear characteristics of *revision* rather than update, as viewed from the perspective of belief change. This mixing of update and revision also seems to be related to work on evolution of *RDFS graphs*,[4] an earlier W3C standard which can be seen as a simple ontology language. While some methods employ revision (Konstantinidis et al., 2007, 2008), others propose both model-based update and a revision-like approach to deal with RDFS updates (Gutierrez et al., 2006, 2011). Hence, it seems that in order to deal with ontology updates in general, one needs to provide an operator that seamlessly combines belief revision with belief updates. This constitutes a considerable challenge in

[4] `http://www.w3.org/RDF/`

itself due to the different nature of these change operations and different principles that underlie them.

Furthermore, despite the issues that belief update operators have with performing TBox updates, they still constitute the main basis for updating ABoxes and thus play an important role in ontology updates. Hence, they need to be reconciled with rule update semantics in order to devise a plausible universal hybrid update semantics.

Unfortunately, there are a number of fundamental and seemingly irreconcilable differences between belief and rule updates, witnessed also by the fact that most belief change postulates are incompatible with rule update semantics (Eiter et al., 2002). These differences can be seen at multiple different levels. First, while belief updates are specified by modifying the *models* of a knowledge base, this lacks the expressivity to capture essential dependencies between literals, expressed in rules (Leite and Pereira, 1997). Rule updates thus rely on the *syntactic structure of rules* to determine the stable models after an update. Since ontological axioms do not have such structure, i.e. they have no *heads* and *bodies*, it is difficult to imagine how the ideas underlying rule updates could be applied to update ontologies. This essentially takes out the possibility of merely adapting a rule update semantics to deal with hybrid knowledge bases since the required syntactic structure is absent.

Additionally, in belief updates the truth value of a literal is typically carried *by inertia from its previous state* which is determined by the set of models of the knowledge base before the update. Since most rule update semantics only assign a set of stable models after each update, the *previous state* of a literal may not be well-defined. For instance, no stable model may have existed before the update. Moreover, rule updates *apply inertia to rules* instead of literals, and active rules determine the set of literals that is *justified*, and thus true. This essentially means that given an initial set of models, a change in truth value of one literal has fixed, predetermined consequences when using a belief update operator, while with rule updates it can affect the truth of any other literal, depending on the rules in the original and updating programs. Also, the reason to deactivate a rule may be retracted and the rule may be used to *reinstate* the literal in its head. No such mechanism is present in belief updates.

These fundamental differences easily lead to a clash of intuitions regarding the truth of a literal given a hybrid knowledge base on which we perform both an ontology and a rule update simultaneously.

If we are to ever be able to achieve a universal semantics for updates of hybrid knowledge bases, we need to find a unifying perspective that would embrace both ontology and rule updates, enabling a deeper understanding of all involved methods and principles, and creating room for their cross-fertilisation, ripening and further development. Since an adaptation of rule update semantics to deal with ontology updates does not seem reasonable, as ontologies lack the syntactic structure of rules which these semantics rely on, what remains is to look for *semantic characterisations of rule updates*.

Therefore, in the second half of this thesis, we diverge from our efforts to directly address hybrid updates and instead address this problem. As a main result, we obtain a novel characterisation of update operators that is capable of capturing belief update operators, both model- and formula-based, as well as the historically first causal rejection-based rule update semantics.

1.6 Road Map and Contributions

The remainder of this work is organised as follows:

Chapter 2 – Background: We present the syntax and semantics of first-order logic, description logics, logic programs and MKNF knowledge bases. Then we provide an overview of existing work on belief update principles and operators together with the derived work on ontology updates. Subsequently, we review the syntax-based approaches to rule updates, including a comparison that reveals the distinguishing as well as common properties of these semantics.

Chapter 3 – Dynamic MKNF Knowledge Bases with Static Rules: Based on a given ontology update operator, we define an update semantics for MKNF knowledge bases in which the ontology component can be updated while the rules remain static. We prove some of its basic theoretical properties, namely that it is faithful to the static semantics for MKNF knowledge bases and to the ontology update operator it is based on, and that it enjoys properties such as respect for primacy of new information, immunity to tautological updates or syntax-independence w.r.t. the ontology and its updates.

Chapter 4 – Layered Dynamic MKNF Knowledge Bases: Starting from a realistic scenario that requires the use of hybrid knowledge bases, we set off to devise an update semantics for MKNF knowledge bases that can be divided into a sequence of ontology and rule layers that interact with one another through a rule-based interface. For this purpose we generalise the well-known splitting theorems for logic programs (Lifschitz and Turner, 1994) and use the ideas behind splitting to define a hybrid update semantics using a modular combination of an ontology update operator with a rule update semantics. We prove that the resulting semantics is faithful to the semantics of MKNF knowledge bases as well as to the constituent update semantics and respects primacy of new information. It is also fully compatible with the semantics proposed in Chapter 3. We demonstrate that it is capable of performing non-trivial updates and resolves conflicts in the *expected* manner.

Chapter 5 – Difficulties with Updates of Hybrid Knowledge Bases: In this chapter we are concerned with general semantic issues arising from updates of TBoxes and of hybrid knowledge bases. We show a formal result pinpointing that most model-based update operators do not provide expected results when applied to TBoxes. Then we use generic examples to demonstrate the deep conflicts between belief and rule updates and show how they make it difficult to propose a universal update semantics for hybrid knowledge bases.

Chapter 6 – Belief Updates on SE-Models: Due to the difficulties we encountered, we turn away from updates of MKNF knowledge bases and instead aim at finding a common basis for both belief and rule updates. We show that belief update postulates and operators can be defined over SE-models, a monotonic characterisation of logic programs that is strictly more expressive than stable models. We also prove a counterpart of Katsuno and Mendelzon's representation theorem for such operators. Perhaps surprisingly, we then uncover a serious drawback of the resulting operators when compared to traditional approaches to rule updates.

Chapter 7 – Semantic Characterisations of Rules and Programs: The previous findings motivate our search for richer semantic characterisations of logic programs that are sufficiently expressive to capture fundamental properties of traditional rule update semantics. We first pinpoint the exact expressivity of SE-models w.r.t. single rules and propose a novel monotonic characterisation of rules, dubbed *RE-models*, which can distinguish important classes of rules that are indistinguishable using SE-models. We propose to view a program as the *set of sets of SE- or RE-models of its rules*, define the corresponding notions of program equivalence and entailment and compare them in terms of their strength.

Chapter 8 – Exception-Based Updates: Viewing a program as the set of sets of RE-models of its rules, we introduce an update framework based on introducing new models – *exceptions* – to the sets of models of rules in the original program. We show that this way we are able to capture the historically first causal rejection-based rule update semantics, enabling us to shed new light on the problem of *state condensing*. We extensively examine the semantic properties of exception-based rule update operators w.r.t. different notions of program entailment and equivalence. Finally, we show that exception-based operators operators can capture a wide range of belief update operators, creating an important bridge between traditional approaches to rule updates and a variety of belief update operators.

Chapter 9 – Conclusions and Future Directions: In the final chapter we summarise the contributions of this thesis and discuss desirable future developments.

Appendices A, B, C, D, E and F – Proofs: Here we present the proofs of theoretical results from Chapters 2, 3, 4, 6, 7 and 8, respectively.

The main contributions of this thesis are as follows:

- We consider ontology updates in the presence of static rules and propose a semantics that can handle them, parametrised by an ontology update operator;
- Motivated by a real-world application, we tackle updates of hybrid knowledge bases that can be split into a sequence of ontology and rule layers, and propose a semantics for it, parametrised by an ontology update operator and a rule update semantics;
- We formally pinpoint why belief update operators yield counterintuitive results when used to update TBoxes (as shown in Example 1.8 for the case of Winslett's operator);
- We illustrate the overwhelming difficulties with defining a universal update semantics for hybrid knowledge bases due to the clash of intuitions underlying belief update operators and rule update semantics;
- We show that SE-models, an expressive monotonic characterisation of logic programs, can be used as a semantic foundation for rule update operators. Nevertheless, the resulting operators suffer from severe disadvantages when compared to traditional syntax-based rule update semantics;
- We propose richer semantic characterisations of logic programs and examine the notions of program equivalence and entailment that follow from them;
- We propose a novel method for defining update operators and show that it can capture model-based as well as formula-based belief update operators and also the historically first causal rejection-based rule update semantics, thus bridging the syntax-based approach to rule updates with a semantic one.

Depending on the reader's preferences and goals, certain parts of this thesis may be more relevant than others. Chapter 2 sets the stage for the following chapters by defining the necessary concepts and providing an overview of existing work on ontology and rule updates. Some parts of it may be skipped if the reader is already familiar with the respective formalisms.

Chapters 3, 4 and 5 form Part II in which we address updates of MKNF knowledge bases. Preliminary versions of some results presented in these chapters have been published in (Slota and Leite, 2010a,b; Slota et al., 2011).

Chapters 6, 7 and 8 form Part III and are concerned with finding semantic characterisations of rule update semantics. Some parts of these chapters have been published in (Slota and Leite, 2010c, 2011, 2012a,b).

Parts II and III are reasonably independent of one another. They can be read in any order and any of them may be skipped without compromising comprehensibility of the other one. This leaves us with three recommended reading paths:

$$1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4 \longrightarrow 5 \longrightarrow 9$$
$$1 \longrightarrow 2 \longrightarrow 6 \longrightarrow 7 \longrightarrow 8 \longrightarrow 9$$
$$1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4 \longrightarrow 5 \longrightarrow 6 \longrightarrow 7 \longrightarrow 8 \longrightarrow 9$$

2

Background

WITH AN INNOCENT SMILE, AMID A LECTURE ON MATHEMATICS...
Now we are going to introduce a few new concepts so that I can express myself more easily and you have a harder time understanding what I'm talking about.

Zbyněk Kubáček
Slovak mathematician, music conductor & very memorable lecturer

In this chapter we introduce the standard syntax and semantics of first-order logic (Section 2.1), description logics (Section 2.2), answer set programming (Section 2.3) and MKNF knowledge bases (Section 2.4).

Subsequently, starting on page 30, we provide an overview of belief update postulates and operators (Section 2.5), their generalisations to first-order logic (Section 2.6), and particular approaches to ontology updates which are based on them (Section 2.7). Some of the terminology we introduce in these sections is specific to this thesis.

Finally, in Section 2.8 we provide a historical overview of many approaches to rule updates and discuss both the differences and similarities between them. Some of the examples and results provided here are novel and may be of interest on their own account.

2.1 First-Order Logic

We summarise the syntax and semantics of standard function-free first-order logic which forms the basis for representing both ontological and rule-based knowledge.

The signature of a first-order language consists of disjoint non-empty sets of *constant* and *predicate symbols* $\mathcal{C}$ and $\mathcal{P}$. Each predicate symbol $P \in \mathcal{P}$ has an associated natural number called its *arity*. Apart from the symbols in the signature, first-order formulae are formed using the following logical symbols:

- a fixed countably infinite set of *variables*, disjoint from $\mathcal{C}$ and $\mathcal{P}$;
- the *equality symbol* '$\approx$';
- the *logical connectives* '$\neg$' and '$\wedge$';
- the *quantifier* '$\exists$' and
- the *auxiliary parenthesis* '(' and ')'.

We denote the set of predicate symbols $\mathcal{P}$ together with the equality symbol $\approx$ by $\mathcal{P}^{\approx}$, i.e. $\mathcal{P}^{\approx} = \mathcal{P} \cup \{ \approx \}$.

The syntax of first-order logic is summarised in the following definition:

Definition 2.1 (First-Order Syntax). A *term* is a constant symbol or a variable.

A *first-order atom* is either of the form $t_1 \approx t_2$ or of the form $P(t_1, t_2, \ldots, t_n)$ where $P \in \mathcal{P}$ is a predicate symbol of arity n and $t_1, t_2, \ldots, t_n$ are terms. We denote the set of all first-order atoms by $\mathcal{A}$.

The set of *first-order formulae* is the smallest set containing

- all first-order atoms;
- $\neg\phi$, $(\phi_1 \wedge \phi_2)$ and $\exists \mathbf{x} : \phi$ for all first-order formulae ϕ, ϕ_1, ϕ_2 and every variable $\mathbf{x}$.

A first-order formula ϕ is *ground* if it contains no variables; *a sentence* if every occurrence of a variable in ϕ is within the scope of a quantifier.

A *first-order theory* is a set of first-order sentences.

In addition, we use the expressions $\bot$, $\top$, $(\phi_1 \vee \phi_2)$, $(\phi_1 \supset \phi_2)$, $(\phi_1 \equiv \phi_2)$ and $\forall \mathbf{x} : \phi$ as shortcuts for the first-order formulae $p \wedge \neg p$, $\neg\bot$, $\neg(\neg\phi_1 \wedge \neg\phi_2)$, $\neg(\phi_1 \wedge \neg\phi_2)$, $(\phi_1 \supset \phi_2) \wedge (\phi_2 \supset \phi_1)$ and $\neg\exists \mathbf{x} : \neg\phi$, respectively, where p is a fixed ground first-order atom. We drop the classification "first-order" as well as the outermost parenthesis in a formula where this does not cause confusion.

Turning to the semantics of first-order logic, we define the standard first-order models as well as a semantics based on interpretations that adopt the standard names assumption and interpret the equality predicate as a congruence relation. This latter semantics is used in the context of MKNF knowledge bases and, as we shall see further on, it is also useful for defining first-order update operators. We conclude this section by pointing out the main differences between these alternative semantics.

Let Δ be a non-empty set called the *universe*. A *first-order interpretation I over Δ* assigns an object $a^I \in \Delta$ to each constant symbol $a \in \mathcal{C}$ and a relation $P^I \subseteq \Delta^n$ to each predicate symbol $P \in \mathcal{P}$ of arity n. We also assume that $d^I = d$ for every $d \in \Delta$ and by $\phi[d/\mathbf{x}]$ we denote the formula obtained from ϕ by replacing every unbound occurrence of variable $\mathbf{x}$ with the object d. Satisfaction of a first-order sentence ϕ and first-order theory $\mathcal{T}$ in I, denoted by $I \models \mathcal{T}$, is defined in Table 2.1.

The semantics of first-order sentences and theories is defined as follows:

Definition 2.2 (First-Order Semantics). Let $\mathcal{T}$ be a first-order theory. A first-order interpretation I is a *first-order model of* $\mathcal{T}$ if $I \models \mathcal{T}$. We denote the set of all first-order models of $\mathcal{T}$ by $[\![\mathcal{T}]\!]_{\mathsf{FO}}$.

Given two first-order theories $\mathcal{T}$, $\mathcal{S}$, we say that $\mathcal{T}$ *first-order entails* $\mathcal{S}$, denoted by $\mathcal{T} \models_{\mathsf{FO}} \mathcal{S}$, if $[\![\mathcal{T}]\!]_{\mathsf{FO}} \subseteq [\![\mathcal{S}]\!]_{\mathsf{FO}}$, and that $\mathcal{T}$ *is first-order equivalent to* $\mathcal{S}$, denoted by $\mathcal{T} \equiv_{\mathsf{FO}} \mathcal{S}$, if $[\![\mathcal{T}]\!]_{\mathsf{FO}} = [\![\mathcal{S}]\!]_{\mathsf{FO}}$.

The first-order models, set of first-order models, first-order entailment and first-order equivalence are generalised to first-order sentences by treating every sentence ϕ as the theory $\{ \phi \}$.

Table 2.1: Standard satisfaction of first-order sentences and theories

	Sentences (inductively)	
$I \models a \approx b$	if and only if	$a^I = b^I$
$I \models P(a_1, a_2, \ldots, a_n)$	if and only if	$(a_1^I, a_2^I, \ldots, a_n^I) \in P^I$
$I \models \neg\phi$	if and only if	$I \not\models \phi$
$I \models \phi_1 \wedge \phi_2$	if and only if	$I \models \phi_1$ and $I \models \phi_2$
$I \models \exists \mathbf{x} : \phi$	if and only if	$I \models \phi[d/\mathbf{x}]$ for some $d \in \Delta$
	Theories	
$I \models \mathcal{T}$	if and only if	$I \models \phi$ for all $\phi \in \mathcal{T}$

The alternative semantics, adopted by Motik and Rosati (2007, 2010) in the context of MKNF knowledge bases, consists of imposing the standard names assumption for compatibility with Logic Programs and interpreting the equality predicate as a congruence relation for compatibility with Description Logics (Fitting, 1996). More formally, in this semantics

1. we assume that the set of constant symbols $\mathcal{C}$ is infinite;
2. we consider only Herbrand interpretations in the semantics, i.e. interpretations I over the universe $\mathcal{C}$ such that $a^I = a$ for every constant symbol $a \in \mathcal{C}$;
3. we allow the equality predicate $\approx$ to be interpreted by an arbitrary congruence relation on $\mathcal{C}$ that allows for replacement of equals by equals, i.e. for every interpretation I,
   i. $\approx^I$ is a reflexive, symmetric and transitive relation on $\mathcal{C}$;
   ii. for every $n \in \mathbb{N}$, every predicate symbol $P \in \mathcal{P}$ of arity n and all constant symbols $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n \in \mathcal{C}$ such that $a_k \approx^I b_k$ for all $k \in \{1, 2, \ldots, n\}$,
   $$(a_1, a_2, \ldots, a_n) \in P^I \qquad \text{if and only if} \qquad (b_1, b_2, \ldots, b_n) \in P^I \ .$$

We denote the set of all interpretations that satisfy the above conditions by $\mathcal{I}$.

Technically, most members of $\mathcal{I}$ are not first-order interpretations as we defined them previously because they may interpret the equality predicate by other equivalence relations than true equality. In what follows we simply call the members of $\mathcal{I}$ *interpretations*, without the classification *first-order*. We only use this classification when we specifically need to refer to standard first-order interpretations as defined previously.

Satisfaction of first-order sentences and theories in an interpretation $I \in \mathcal{I}$ is defined very similarly to first-order satisfaction, the only difference being the satisfaction of equality atoms. It is summarised in Table 2.2.

We define the *models* and, based on them, also *entailment* and *equivalence* between first-order theories and sentences, as follows:

Definition 2.3 (Standard Names Semantics of First-Order Theories)**.** Let $\mathcal{T}$ be a first-order theory and $I \in \mathcal{I}$ an interpretation. We say that I *is a model of* $\mathcal{T}$ if $I \models \mathcal{T}$ and denote the set of all models of $\mathcal{T}$ by $[\![\mathcal{T}]\!]$.

Given two first-order theories $\mathcal{T}$, $\mathcal{S}$, we say that $\mathcal{T}$ *entails* $\mathcal{S}$, denoted by $\mathcal{T} \models \mathcal{S}$, if $[\![\mathcal{T}]\!] \subseteq [\![\mathcal{S}]\!]$, and that $\mathcal{T}$ *is equivalent to* $\mathcal{S}$, denoted by $\mathcal{T} \equiv \mathcal{S}$, if $[\![\mathcal{T}]\!] = [\![\mathcal{S}]\!]$. We also say that $\mathcal{T}$ is *satisfiable* if $[\![\mathcal{T}]\!]$ is non-empty.

The models, set of models, entailment and equivalence are generalised to first-order sentences by treating every sentence ϕ as the theory $\{\phi\}$.

Table 2.2: Satisfaction of first-order sentences and theories in $I \in \mathfrak{I}$

	Sentences (inductively)	
$I \models a \approx b$	if and only if	$a \approx^I b$
$I \models P(a_1, a_2, \ldots, a_n)$	if and only if	$(a_1^I, a_2^I, \ldots, a_n^I) \in P^I$
$I \models \neg\phi$	if and only if	$I \not\models \phi$
$I \models \phi_1 \wedge \phi_2$	if and only if	$I \models \phi_1$ and $I \models \phi_2$
$I \models \exists \mathbf{x} : \phi$	if and only if	$I \models \phi[a/\mathbf{x}]$ for some $a \in \mathfrak{C}$
	Theories	
$I \models \mathcal{T}$	if and only if	$I \models \phi$ for all $\phi \in \mathcal{T}$

The basic property of the defined models is that standard first-order entailment (and equivalence) between finite first-order theories can be equivalently expressed in terms of entailment on models from $\mathfrak{I}$. This follows from the results of Fitting (1996) since in case of a finite theory, there always remains an infinite supply of constant symbols that are not used in it. Formally:

Proposition 2.4. *Let $\mathcal{T}$, $\mathcal{S}$ be finite first-order theories. Then $\mathcal{T} \models_{\mathsf{FO}} \mathcal{S}$ if and only if $\mathcal{T} \models \mathcal{S}$.*

Proof. Follows from Theorems 5.9.4 and 9.3.9 in (Fitting, 1996). □

But in case of infinite first-order theories this result is preserved only in one direction: standard first-order entailment implies entailment on models from $\mathfrak{I}$. The converse implication does not in general hold because an infinite theory can use up too many constant symbols from $\mathfrak{C}$.

Proposition 2.5. *Let $\mathcal{T}$, $\mathcal{S}$ be first-order theories. Then $\mathcal{T} \models_{\mathsf{FO}} \mathcal{S}$ implies $\mathcal{T} \models \mathcal{S}$ but the converse implication does not in general hold.*

Proof. See Appendix A, page 167. □

This means that care needs to be taken when we manipulate infinite first-order theories and consider their *models* instead of the standard *first-order models*.

2.2 Description Logics

Fragments of classical first-order logic with decidable reasoning tasks are typically called *Description Logics* (DLs) (Baader et al., 2007). Unless stated otherwise, in this thesis we do not constrain ourselves to a specific DL for representing ontologies. The assumption taken in the theoretical developments is that the ontology language is a syntactic variant of a fragment of function-free first-order logic, covering also cases when the fragment would not normally be considered a description logic. We assume that its semantics is given by translation to a finite first-order theory. Such translations are known for most DLs (Baader et al., 2007).

Definition 2.6 (DL Semantics by Translation)**.** Let ϕ be a DL axiom. By $\kappa(\phi)$ we denote some first-order sentences that *semantically characterises* ϕ.

An *ontology* is a finite set of DL axioms. For a DL ontology $\mathcal{O}$, $\kappa(\mathcal{O}) = \{ \kappa(\phi) \mid \phi \in \mathcal{O} \}$.

Given two ontologies $\mathcal{O}$, $\mathcal{O}'$, we say that $\mathcal{O}$ *entails* $\mathcal{O}'$, denoted by $\mathcal{O} \models \mathcal{O}'$, if $\kappa(\mathcal{O}) \models_{\mathsf{FO}} \kappa(\mathcal{O}')$, and that $\mathcal{O}$ *is equivalent to* $\mathcal{O}'$, denoted by $\mathcal{O} \equiv \mathcal{O}'$, if $\kappa(\mathcal{O}) \equiv_{\mathsf{FO}} \kappa(\mathcal{O}')$.

Given an ontology $\mathcal{O}$ and a first-order sentence ϕ, we say that $\mathcal{O}$ *entails* ϕ, denoted by $\mathcal{O} \models \phi$, if $\kappa(\mathcal{O}) \models_{\mathsf{FO}} \phi$.

Table 2.3: Interpretation of $\mathcal{ALCIO}$ expressions

Expression	Notation	Interpretation
Role expressions		
Role name	R	R^I
Inverse role	R^-	$\{ (d,e) \in \Delta \times \Delta \mid (e,d) \in R^I \}$
Concept expressions		
Universal concept	$\top$	Δ
Empty concept	$\bot$	$\emptyset$
Concept name	A	A^I
Concept intersection	$C \sqcap D$	$C^I \cap D^I$
Concept union	$C \sqcup D$	$C^I \cup D^I$
Concept negation	$\neg C$	$\Delta \setminus C^I$
Universal restriction	$\forall R.C$	$\{ d \in \Delta \mid \forall e \in \Delta : (d,e) \in R^I \Longrightarrow e \in C^I \}$
Existential restriction	$\exists R.C$	$\{ d \in \Delta \mid \exists e \in \Delta : (d,e) \in R^I \wedge e \in C^I \}$
Nominal	$\{ a \}$	$\{ a^I \}$

For the sake of completeness, in what follows we briefly present the syntax and direct semantics of the DL $\mathcal{ALCIO}$, an extension of $\mathcal{ALC}$ with inverse roles ($\mathcal{I}$) and nominals ($\mathcal{O}$) that is used in examples throughout this work. A reader interested in further details about DLs, reasoning tasks and their computational complexity can refer to (Baader et al., 2007).

A DL signature consists of pairwise disjoint sets of *individual names*, *concept names* and *role names*. These represent objects, groups of objects and binary relations between objects, respectively. Given these three sets, complex role and concept expressions can be formed and used to form ontological axioms. These are divided in two groups: *TBox axioms* with assertions about the terminology and *ABox axioms* with assertions about individuals. The formal definitions in case of $\mathcal{ALCIO}$ are as follows:

Definition 2.7 ($\mathcal{ALCIO}$ Syntax)**.** An $\mathcal{ALCIO}$ *role expression* is either a role name R or its inverse, denoted by R^-.

The set of $\mathcal{ALCIO}$ *concept expressions* is the smallest set containing $\top$, $\bot$, all concept names and the expressions $\neg C$, $C \sqcap D$, $C \sqcup D$, $\forall R.C$, $\exists R.C$ and $\{ a \}$ for all concept expressions C, D, all role expressions R and every individual name a.

An $\mathcal{ALCIO}$ *TBox* is a finite set of axioms of the forms $C \sqsubseteq D$ and $C \equiv D$ where C, D are concept expressions. A TBox is *acyclic* if it contains only axioms of the form $A \equiv C$ such that A is a concept name, C is a concept expression that does not refer directly or indirectly to A, and A does not occur in on the left-hand side of any other axiom in $\mathcal{T}$.

An $\mathcal{ALCIO}$ *ABox* is a finite set of assertions of the forms $C(a)$, $R(a,b)$, $\neg R(a,b)$, $a \approx b$ and $a \not\approx b$ where C is a concept expression, R is a role name and a, b are individual names.

An $\mathcal{ALCIO}$ *ontology* is a union of an $\mathcal{ALCIO}$ TBox and of an $\mathcal{ALCIO}$ ABox.

The semantics of most DLs is very similar to that of first-order logic. A *DL interpretation over a universe* Δ is a mapping I that assigns an object $a^I \in \Delta$ to every individual name a, a set of objects $A^I \subseteq \Delta$ to every concept name A and a set of pairs of objects $R^I \subseteq \Delta \times \Delta$ to every role name R. This mapping is then generalised to concept and role expressions and satisfaction of ontologies in a DL interpretation I is defined. Tables 2.3 and 2.4 summarise these definitions for the case of $\mathcal{ALCIO}$.

Table 2.4: Satisfaction of $\mathcal{ALCIO}$ axioms and ontologies

TBox axioms		
$I \models C \sqsubseteq D$	if and only if	$C^I \subseteq D^I$
$I \models C \equiv D$	if and only if	$C^I = D^I$
ABox axioms		
$I \models C(a)$	if and only if	$a^I \in C^I$
$I \models R(a, b)$	if and only if	$(a^I, b^I) \in R^I$
$I \models \neg R(a, b)$	if and only if	$(a^I, b^I) \notin R^I$
$I \models a \approx b$	if and only if	$a^I = b^I$
$I \models a \not\approx b$	if and only if	$a^I \neq b^I$
Ontologies		
$I \models \mathcal{O}$	if and only if	$I \models \phi$ for all $\phi \in \mathcal{O}$

Based on these definitions, entailment and equivalence between $\mathcal{ALCIO}$ ontologies can be defined as follows:

Definition 2.8 ($\mathcal{ALCIO}$ Semantics)**.** Let $\mathcal{O}$ be an $\mathcal{ALCIO}$ ontology. A DL interpretation I is a *DL model of* $\mathcal{O}$ if $I \models \mathcal{O}$. We denote the set of all DL models of $\mathcal{O}$ by $[\![\mathcal{O}]\!]_{\mathsf{DL}}$.

Given two ontologies $\mathcal{O}$, $\mathcal{O}'$, we say that $\mathcal{O}$ *DL entails* $\mathcal{O}'$, denoted by $\mathcal{O} \models_{\mathsf{DL}} \mathcal{O}'$, if $[\![\mathcal{O}]\!]_{\mathsf{DL}} \subseteq [\![\mathcal{O}']\!]_{\mathsf{DL}}$, and that $\mathcal{O}$ *is DL equivalent to* $\mathcal{O}'$, denoted by $\mathcal{O} \equiv_{\mathsf{DL}} \mathcal{O}'$, if $[\![\mathcal{O}]\!]_{\mathsf{DL}} = [\![\mathcal{O}']\!]_{\mathsf{DL}}$.

2.3 Answer Set Programming

Similarly as Description Logics, Logic Programming has its roots in classical first-order logic. However, Logic Programs diverge from first-order semantics by adopting the Closed World Assumption and allowing for non-monotonic inferences. In what follows we introduce the syntax of extended logic programs that allow for both disjunction and default negation in heads of rules. Then we define the *stable models* of such programs, also referred to as *answer sets* in the literature (Gelfond and Lifschitz, 1988, 1991; Lifschitz and Woo, 1992; Baral, 2003).

Syntactically, logic programs are built from *first-order atoms without equality*, as defined in Section 2.1. An *objective literal* is a first-order atom $p = P(t_1, t_2, \dots, t_n)$ or its (strong) negation $\neg p$. We denote the set of all objective literals by $\mathcal{L}$ and use the following notation to refer to complementary objective literals: $\overline{p} = \neg p$ and $\overline{\neg p} = p$ for all atoms p. A *default literal* is an objective literal preceded by $\sim$ denoting *default negation*. A *literal* is either an objective literal or a default literal. As a convention, double default negation is absorbed, so that $\sim\sim l$ denotes the objective literal l. Given a set of literals S, we introduce the following notation:

$$S^+ = \{ l \in \mathcal{L} \mid l \in S \} \enspace , \qquad S^- = \{ l \in \mathcal{L} \mid \sim l \in S \} \enspace , \qquad \sim S = \{ \sim L \mid L \in S \} \enspace .$$

Definition 2.9 (Logic Program Syntax)**.** A *rule* is a pair of finite sets of literals $\pi = (H(\pi), B(\pi))$. We say that $H(\pi)$ is the *head of* π and $B(\pi)$ is the *body of* π. Usually, for convenience, we write π as

$$H(\pi)^+ ; \sim H(\pi)^- \leftarrow B(\pi)^+, \sim B(\pi)^- .$$

We also say that $H(\pi)^+$ is the *positive head of* π, $H(\pi)^-$ the *negative head of* π, $B(\pi)^+$ the

Table 2.5: Satisfaction of ground literals, rules and programs

$J \models l$	if and only if	$l \in J$
$J \models \sim l$	if and only if	$l \notin J$
$J \models S$	if and only if	$J \models L$ for all $L \in S$
$J \models \pi$	if and only if	$\exists L \in B(\pi) : J \not\models L$ or $\exists L \in H(\pi) : J \models L$
$J \models \mathcal{P}$	if and only if	$J \models \pi$ for all $\pi \in \mathcal{P}$

positive body of π and $B(\pi)^-$ the *negative body of* π.

A rule is called *ground* if it does not contain variables; *positive* if it does not contain any default literal; *non-disjunctive* if its head contains at most one literal; *a fact* if its head contains exactly one literal and its body is empty. The *grounding* of a rule π is the set of rules $\mathsf{ground}(\pi)$ obtained by replacing in π all variables with constant symbols from $\mathcal{C}$ in all possible ways.

A *program* is a set of rules. A program is *ground* if all its rules are ground; *positive* if all its rules are positive; *non-disjunctive* if all its rules are non-disjunctive. The grounding of a program $\mathcal{P}$ is defined as $\mathsf{ground}(\mathcal{P}) = \bigcup_{\pi \in \mathcal{P}} \mathsf{ground}(\pi)$.

We define the class of *acyclic programs* in the standard way using level mappings Apt and Bezem (1991).

Definition 2.10 (Apt and Bezem (1991))**.** A *level mapping* is a function ℓ that assigns a natural number to every ground objective literal and is extended to ground default literals and sets of ground literals by putting $\ell(\sim L) = \ell(L)$ and $\ell(S) = \max \{ \ell(L) \mid L \in S \}$.

We say that a program $\mathcal{P}$ is *acyclic* if there exists a level mapping ℓ such that for every rule $\pi \in \mathsf{ground}(\mathcal{P})$ it holds that $\ell(H(\pi)) > \ell(B(\pi))$.

The stable models of logic programs can be obtained from first-order semantics by considering interpretations in which all constant symbols are interpreted by themselves, and by interpreting each ground atom p separately of (though still consistently with) its strong negation $\neg p$. Each such interpretation naturally corresponds to a consistent subset of the set of all ground objective literals $\mathcal{L}_\mathsf{G}$. A stable model is then an interpretation that satisfies all rules and at the same time can be fully derived using the rules of the program assuming that literals not present in it are false by default.

More formally, an *ASP interpretation* is a subset of $\mathcal{L}_\mathsf{G}$ that does not contain both l and $\bar{l}$ for any ground objective literal l. Satisfaction of ground programs is obtained by treating rules as classical implications – Table 2.5 defines satisfaction of ground literals l and $\sim l$, a set of ground literals S, a ground rule π and a ground program $\mathcal{P}$ in an ASP interpretation $J \subseteq \mathcal{L}_\mathsf{G}$. We also say that J is a *model of* $\mathcal{P}$ if $J \models \mathcal{P}$.

The stable models are established by the following definition:

Definition 2.11 (Stable Model)**.** Let $\mathcal{P}$ be a ground program. An ASP interpretation J is a *stable model of* $\mathcal{P}$ if and only if J is a subset-minimal model of the *reduct of* $\mathcal{P}$ *relative to* J

$$\mathcal{P}^J = \left\{ H(\pi)^+ \leftarrow B(\pi)^+. \;\middle|\; \pi \in \mathcal{P} \wedge J \not\models (\sim H(\pi)^- \leftarrow \sim B(\pi)^-.) \right\} .$$

The stable models of a non-ground program $\mathcal{P}$ are the stable models of $\mathsf{ground}(\mathcal{P})$. The set of all stable models of a program $\mathcal{P}$ is denoted by $[\![\mathcal{P}]\!]_{\mathsf{SM}}$.

A program is *consistent* if it has a model; *coherent* if it has a stable model.

Note that, unlike in (Gelfond and Lifschitz, 1991), we do not allow an inconsistent program to have a stable model.

2.4 MKNF Knowledge Bases

We first introduce the logic of Minimal Knowledge and Negation as Failure (MKNF) (Lifschitz, 1991) that forms the logical basis of (Hybrid) MKNF Knowledge Bases (Motik and Rosati, 2007, 2010). Subsequently, we define the syntax of MKNF knowledge bases and provide semantics for them by translation to MKNF theories.

2.4.1 Minimal Knowledge and Negation as Failure

Syntactically, MKNF is an extension of first-order logic (c.f. Section 2.1) with two modal operators: **K** and **not**.

Definition 2.12 (MKNF Syntax)**.** The set of *MKNF formulae* is the smallest set containing

- all first-order atoms;
- $\neg\phi$, $(\phi_1 \wedge \phi_2)$ and $\exists \mathbf{x} : \phi$ for all MKNF formulae ϕ, ϕ_1, ϕ_2 and every variable $\mathbf{x}$;
- $\mathbf{K}\,\phi$ and $\mathbf{not}\,\phi$ for all MKNF formulae ϕ.

An MKNF formula ϕ is *ground* if it contains no variables; *a sentence* if every occurrence of a variable in ϕ is within the scope of a quantifier. An MKNF formula of the form $\mathbf{K}\,\phi$ is called a *modal* **K**-*atom*, and a formula of the form $\mathbf{not}\,\phi$ is called a *modal* **not**-*atom*; collectively, modal **K**- and **not**-atoms are called *modal atoms*.

An *MKNF theory* is a set of MKNF sentences.

Similarly as with first-order formulae, we use the expressions $\bot$, $\top$, $(\phi_1 \vee \phi_2)$, $(\phi_1 \supset \phi_2)$, $(\phi_1 \equiv \phi_2)$ and $\forall \mathbf{x} : \phi$ as shortcuts for the MKNF formulae $p \wedge \neg p$, $\neg\bot$, $\neg(\neg\phi_1 \wedge \neg\phi_2)$, $\neg(\phi_1 \wedge \neg\phi_2)$, $(\phi_1 \supset \phi_2) \wedge (\phi_2 \supset \phi_1)$ and $\neg\exists \mathbf{x} : \neg\phi$, respectively, where p is a fixed ground first-order atom. We drop the classification "MKNF" as well as the outermost parenthesis in a formula where this does not cause confusion.

We adopt the semantics for MKNF theories that was motivated, introduced and discussed in (Motik and Rosati, 2007, 2010). It is based on using the set of interpretations $\mathfrak{I}$ that adopt the standard names assumption (see Section 2.1 for a definition) to interpret the first-order connectives and providing additional semantic structures for interpreting the modal operators. An *MKNF structure* is a triple $(I, \mathcal{M}, \mathcal{N})$ where $I \in \mathfrak{I}$ and $\mathcal{M}, \mathcal{N} \subseteq \mathfrak{I}$.[1] Intuitively, the first component is used to interpret the first-order parts of an MKNF sentence while the other two components interpret the **K** and **not** modalities, respectively. By $\phi[a/\mathbf{x}]$ we denote the formula obtained from ϕ by replacing every unbound occurrence of variable $\mathbf{x}$ with the constant symbol a. The satisfaction of a ground first-order atom p, an MKNF sentence ϕ and MKNF theory $\mathcal{T}$ in an MKNF structure $(I, \mathcal{M}, \mathcal{N})$ and in any $\mathcal{M} \subseteq \mathfrak{I}$ is defined in Table 2.6.

An *MKNF interpretation* $\mathcal{M}$ is a non-empty subset of $\mathfrak{I}$.[2] By $\mathfrak{M} = 2^{\mathfrak{I}}$ we denote the set of all MKNF interpretations together with the empty set. The semantics of MKNF sentences and theories is defined as follows:

Definition 2.13 (MKNF Semantics)**.** Let $\mathcal{T}$ be an MKNF theory. We say that an MKNF interpretation $\mathcal{M}$ is

- an *S5 model of* $\mathcal{T}$ if $\mathcal{M} \models \mathcal{T}$;

[1]Differently from (Motik and Rosati, 2010), we allow for empty $\mathcal{M}, \mathcal{N}$ in this definition as later on it will be useful to have satisfaction defined for this marginal case.

[2]Notice that if $\mathcal{M}$ is empty, then it vacuously holds that $\mathcal{M} \models \phi$ for all sentences ϕ. For this reason, and in accordance with (Motik and Rosati, 2010), the empty set is not considered an MKNF interpretation.

Table 2.6: Satisfaction of MKNF sentences and theories

	Sentences (inductively)	
$(I, \mathcal{M}, \mathcal{N}) \models p$	if and only if	$I \models p$
$(I, \mathcal{M}, \mathcal{N}) \models \neg\phi$	if and only if	$(I, \mathcal{M}, \mathcal{N}) \not\models \phi$
$(I, \mathcal{M}, \mathcal{N}) \models \phi_1 \land \phi_2$	if and only if	$(I, \mathcal{M}, \mathcal{N}) \models \phi_1$ and $(I, \mathcal{M}, \mathcal{N}) \models \phi_2$
$(I, \mathcal{M}, \mathcal{N}) \models \exists \mathbf{x} : \phi$	if and only if	$(I, \mathcal{M}, \mathcal{N}) \models \phi[a/\mathbf{x}]$ for some $a \in \mathcal{C}$
$(I, \mathcal{M}, \mathcal{N}) \models \mathbf{K}\,\phi$	if and only if	$(J, \mathcal{M}, \mathcal{N}) \models \phi$ for all $J \in \mathcal{M}$
$(I, \mathcal{M}, \mathcal{N}) \models \mathbf{not}\,\phi$	if and only if	$(J, \mathcal{M}, \mathcal{N}) \not\models \phi$ for some $J \in \mathcal{N}$
	Theories	
$(I, \mathcal{M}, \mathcal{N}) \models \mathcal{T}$	if and only if	$(I, \mathcal{M}, \mathcal{N}) \models \phi$ for all $\phi \in \mathcal{T}$
	Satisfaction in $\mathcal{M} \subseteq \mathfrak{I}$	
$\mathcal{M} \models \phi$	if and only if	$(I, \mathcal{M}, \mathcal{M}) \models \phi$ for all $I \in \mathcal{M}$
$\mathcal{M} \models \mathcal{T}$	if and only if	$(I, \mathcal{M}, \mathcal{M}) \models \mathcal{T}$ for all $I \in \mathcal{M}$

- an *MKNF model of $\mathcal{T}$* if $\mathcal{M}$ is an S5 model of $\mathcal{T}$ and for every MKNF interpretation $\mathcal{M}' \supsetneq \mathcal{M}$ there is some $I' \in \mathcal{M}'$ such that $(I', \mathcal{M}', \mathcal{M}) \not\models \mathcal{T}$.

Given two MKNF theories $\mathcal{T}, \mathcal{S}$, we say that $\mathcal{T}$ *MKNF entails* $\mathcal{S}$, denoted by $\mathcal{T} \models_{\mathsf{MKNF}} \mathcal{S}$, if $\mathcal{M} \models \mathcal{S}$ for every MKNF model $\mathcal{M}$ of $\mathcal{T}$, and that $\mathcal{T}$ *is MKNF equivalent to* $\mathcal{S}$, denoted by $\mathcal{T} \equiv_{\mathsf{MKNF}} \mathcal{S}$, if $\mathcal{T} \models_{\mathsf{MKNF}} \mathcal{S}$ and $\mathcal{S} \models_{\mathsf{MKNF}} \mathcal{T}$. We also say that $\mathcal{T}$ is *MKNF satisfiable* if $\mathcal{T}$ has an MKNF model.

The S5 and MKNF models, MKNF entailment and MKNF equivalence are generalised to MKNF sentences by treating every MKNF sentence ϕ as the MKNF theory $\{ \phi \}$.

2.4.2 Syntax and Semantics of MKNF Knowledge Bases

MKNF knowledge bases (Motik and Rosati, 2010) consist of two components over a shared signature – an ontology $\mathcal{O}$ and a program $\mathcal{P}$ – and their semantics is given by translation to an MKNF theory. In the following we introduce the syntax and semantics of MKNF knowledge bases with the following deviations from (Motik and Rosati, 2010):

- We allow for default negation in heads of MKNF rules due to its importance in rule update semantics based on causal rejection (e.g. (Leite and Pereira, 1997; Alferes et al., 2000, 2005); c.f. Section 2.8 for an overview). As we shall demonstrate, this renders the resulting formalism strictly more expressive than MKNF knowledge bases as defined in (Motik and Rosati, 2010), though no expressivity is added when we constrain ourselves to non-disjunctive rules – the typical case tackled by existing rule update semantics.
- We do not directly include the **K** and **not** modalities in MKNF rules; instead, they are introduced upon translation to an MKNF theory. We can afford to do this because we do not consider MKNF$^+$ knowledge bases that allow for non-modal rule components. Note that, as shown in (Motik and Rosati, 2010), every MKNF$^+$ knowledge base can be equivalently translated to an MKNF knowledge base over a large enough generalised atom base.[3]
- The translation function κ (denoted by π in (Motik and Rosati, 2010)) is overridden to also accept atoms, literals and sets of literals and produces an MKNF theory

[3] A *generalised atom base* is formally defined in the following paragraph.

instead of an MKNF sentence. The reason for the latter is that we do not assume an MKNF program to be finite so that we can immediately deal with infinite ground programs that result from grounding a finite but non-ground program (the same is actually done in (Motik and Rosati, 2010) from Section 4 onwards).

Similarly as in Section 2.1, we assume a function-free first-order signature consisting of disjoint sets of constant and predicate symbols $\mathcal{C}$ and $\mathcal{P}$. A *generalised atom* is a first-order formula. A generalised atom is *ground* if it is a sentence.[4] A *generalised atom base* $\mathcal{B}$ is a set of generalised atoms such that $\xi \in \mathcal{B}$ implies $\xi_\mathsf{G} \in \mathcal{B}$ whenever ξ_G is obtained from ξ by replacing all its free variables with constants from $\mathcal{C}$.

A *generalised default literal* is a generalised atom preceded by $\sim$. A *generalised literal* is either a generalised atom or a generalised default literal. As a convention, double default negation is absorbed, so that $\sim\sim\xi$ denotes the generalised atom ξ. Given a set of generalised literals S, we introduce the following notation:

$$S^+ = \{\, \xi \in \mathcal{B} \mid \xi \in S \,\} \quad , \qquad S^- = \{\, \xi \in \mathcal{B} \mid {\sim}\xi \in S \,\} \quad , \qquad {\sim}S = \{\, {\sim}L \mid L \in S \,\} \quad .$$

Definition 2.14 (Syntax of MKNF Knowledge Bases)**.** An *MKNF rule* is a pair of finite sets of generalised literals $\pi = (H(\pi), B(\pi))$. We say that $H(\pi)$ is the *head of* π and $B(\pi)$ is the *body of* π. Usually, for convenience, we write π as

$$H(\pi)^+; {\sim}H(\pi)^- \leftarrow B(\pi)^+, {\sim}B(\pi)^-.$$

We also say that $H(\pi)^+$ is the *positive head of* π, $H(\pi)^-$ the *negative head of* π, $B(\pi)^+$ the *positive body of* π and $B(\pi)^-$ the *negative body of* π.

An MKNF rule is called *ground* if it contains only ground generalised atoms; *positive* if it does not contain any generalised default literal; *non-disjunctive* if its head contains at most one generalised literal; *a fact* if its head contains exactly one generalised literal and its body is empty. The *grounding* of a rule π is the set of rules $\mathsf{ground}(\pi)$ obtained by replacing in π all generalised atoms with their groundings in all possible ways.

An *MKNF program* is a set of MKNF rules. An MKNF program is *ground* if all its rules are ground; *positive* if all its rules are positive; *non-disjunctive* if all its rules are non-disjunctive. The grounding of a program $\mathcal{P}$ is defined as $\mathsf{ground}(\mathcal{P}) = \bigcup_{\pi \in \mathcal{P}} \mathsf{ground}(\pi)$.

An *MKNF knowledge base* is a pair $(\mathcal{O}, \mathcal{P})$ where $\mathcal{O}$ is an ontology and $\mathcal{P}$ is an MKNF program. An MKNF knowledge base is *ground* if $\mathcal{P}$ is ground; *positive* if $\mathcal{P}$ is positive; *non-disjunctive* if $\mathcal{P}$ is non-disjunctive. The grounding of an MKNF knowledge base $\mathcal{K}$ is defined as $\mathsf{ground}(\mathcal{K}) = (\mathcal{O}, \mathsf{ground}(\mathcal{P}))$.

The semantics of MKNF knowledge bases is determined by translation into an MKNF theory. To this end we utilise the function κ introduced in Section 2.2 to translate an ontology into a first-order theory and override it to translate every MKNF knowledge base $\mathcal{K}$ to an MKNF theory $\kappa(\mathcal{K})$ as shown in Table 2.7.[5]

The semantics of MKNF knowledge bases is thus defined as follows:

Definition 2.15 (Semantics of MKNF Knowledge Bases)**.** Let $\mathcal{K}$ be an MKNF knowledge base. We say that an MKNF interpretation $\mathcal{M}$ is an *S5 model of* $\mathcal{K}$ if $\mathcal{M}$ is an S5 model of $\kappa(\mathcal{K})$. Similarly, $\mathcal{M}$ is an *MKNF model of* $\mathcal{K}$ if $\mathcal{M}$ is an MKNF model of $\kappa(\mathcal{K})$. MKNF entailment and MKNF equivalence are generalised to MKNF knowledge bases by treating every MKNF knowledge base $\mathcal{K}$ as the MKNF theory $\kappa(\mathcal{K})$.

[4]Note that a ground generalised atom may contain variables bound by a quantifier.

[5]Note that $\bigwedge \emptyset = \top$ and $\bigvee \emptyset = \bot$.

Table 2.7: Transformation of MKNF Knowledge Base to MKNF Theory

Generalised atom ξ	$\kappa(\xi) = \mathbf{K}\,\xi$
Generalised default literal $\sim\xi$	$\kappa(\sim\xi) = \mathbf{not}\,\xi$
Set of generalised literals S	$\kappa(S) = \{\, \kappa(L) \mid L \in S \,\}$
MKNF rule π with vector of free variables $\vec{\mathbf{x}}$	$\kappa(\pi) = \forall \vec{\mathbf{x}} : \bigwedge \kappa(B(\pi)) \supset \bigvee \kappa(H(\pi))$
MKNF program $\mathcal{P}$	$\kappa(\mathcal{P}) = \{\, \kappa(\pi) \mid \pi \in \mathcal{P} \,\}$
MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$	$\kappa(\mathcal{K}) = \{\, \mathbf{K}\,\kappa(\phi) \mid \phi \in \mathcal{O} \,\} \cup \kappa(\mathcal{P})$

At times it is useful to refer to the set of predicate symbols that are mentioned in an ontology, program, MKNF knowledge base or MKNF theory. It can be defined straightforwardly as follows:

Definition 2.16 (Set of Relevant Predicate Symbols)**.** Let ϕ be an MKNF sentence. We inductively define the *set of predicate symbols relevant to* ϕ, denoted by $\mathsf{pr}(\phi)$, as follows:

- If ϕ is a first-order atom $t_1 \approx t_2$, then $\mathsf{pr}(\phi) = \{ \approx \}$;
- If ϕ is a first-order atom $P(t_1, t_2, \dots, t_n)$, then $\mathsf{pr}(\phi) = \{ P \}$;
- If ϕ is of one of the forms $\neg\psi$, $\exists \mathbf{x} : \psi$, $\mathbf{K}\,\psi$ or $\mathbf{not}\,\psi$, then $\mathsf{pr}(\phi) = \mathsf{pr}(\psi)$;
- If ϕ is of the form $(\psi_1 \wedge \psi_2)$, then $\mathsf{pr}(\phi) = \mathsf{pr}(\psi_1) \cup \mathsf{pr}(\psi_2)$.

For an MKNF theory $\mathcal{T}$ and ontology axiom, ontology, literal, rule, program or MKNF knowledge base Ω, $\mathsf{pr}(\mathcal{T}) = \bigcup_{\phi \in \mathcal{T}} \mathsf{pr}(\phi)$ and $\mathsf{pr}(\Omega) = \mathsf{pr}(\kappa(\Omega))$.

Note that this definition assumes that the translation function κ is "well-behaved" w.r.t. ontology axioms in the sense that, apart from the equality predicate $\approx$, it does not introduce predicate symbols that were not present in the argument axiom itself.

2.4.3 Basic Properties

As was shown in (Motik and Rosati, 2010), MKNF knowledge bases are faithful to both the ontology semantics and to the stable models semantics for logic programs. This follows by the same arguments as Proposition 2.4.

Proposition 2.17 (Faithfulness w.r.t. Ontologies (Motik and Rosati, 2010))**.** *Let $\mathcal{O}$ be an ontology and ϕ a first-order sentence. Then $\mathcal{O} \models \phi$ if and only if $(\mathcal{O}, \emptyset) \models_{\mathsf{MKNF}} \phi$.*

Given a large enough generalised atom base (one that contains all first-order atoms without equality as well as their negations), a logic program is just a special case of an MKNF program. Furthermore, every ASP interpretation directly corresponds to an MKNF interpretation and every MKNF interpretation can be approximated by an ASP interpretation as follows:

Definition 2.18 (Correspondence Between ASP and MKNF Interpretations)**.** Let J be an ASP interpretation and $\mathcal{M}$ an MKNF interpretation. The *MKNF interpretation corresponding to J* is $\{\, I \in \mathfrak{I} \mid I \models J \,\}$. The *ASP interpretation corresponding to $\mathcal{M}$* is $\{\, l \in \mathcal{L}_{\mathsf{G}} \mid \mathcal{M} \models l \,\}$.

The standard names assumption enforced on MKNF interpretations is sufficient to conclude the following:

Proposition 2.19 (Faithfulness w.r.t. Stable Models (Motik and Rosati, 2010))**.** *Let $\mathcal{P}$ be a logic program and $\mathcal{K}$ the MKNF knowledge base $(\emptyset, \mathcal{P})$. If J is a stable model of $\mathcal{P}$, then the MKNF interpretation corresponding to J is an MKNF model of $\mathcal{K}$. If $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, then the ASP interpretation corresponding to $\mathcal{M}$ is a stable model of $\mathcal{P}$.*

In other words, the stable models of a logic program $\mathcal{P}$ are in one-to-one correspondence with MKNF models of the MKNF knowledge base $(\emptyset, \mathcal{P})$.

In the remainder of this thesis we will focus only on *ground* MKNF knowledge bases. This simplification is justified by the following lemma which follows from the standard names assumption imposed by MKNF interpretations.

Lemma 2.20 (Motik and Rosati (2010))**.** *The MKNF models of an MKNF knowledge base $\mathcal{K}$ coincide with the MKNF models of its grounding* $\mathsf{ground}(\mathcal{K})$*.*

2.4.4 Notes on Expressivity

As mentioned earlier, MKNF knowledge bases defined above add expressivity to the MKNF knowledge bases defined in (Motik and Rosati, 2010). In particular, it is not difficult to show that without default negation in heads of rules, all MKNF models are subset-maximal:

Proposition 2.21. *Let $\mathcal{K}$ be an MKNF knowledge base without default negation in heads of rules. If $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, then $\mathcal{M}$ is a subset-maximal S5 model of $\mathcal{K}$.*

Proof. See Appendix A, page 172. □

If default negation is allowed in heads of rules, it can be used to generate non-maximal MKNF models. For example, the rule

$$p; \sim p.$$

has two MKNF models $\mathcal{M}_1 = \mathfrak{I}$ and $\mathcal{M}_2 = \{ I \in \mathfrak{I} \mid I \models p \}$, the latter a proper subset of the former. This means that the algorithms and complexity results from (Motik and Rosati, 2010) may not be fully applicable to MKNF knowledge bases as defined here. Nevertheless, our definition is made so general for the sake of uniformity and convenience since this way MKNF knowledge bases are able to capture logic programs with default negation in the head, as introduced in Section 2.3. It is also worth noting that the rule update semantics that rely on default negation in heads of rules consider only non-disjunctive rules and, as shown by the following proposition, no extra expressivity is added by allowing default negation in heads of *non-disjunctive* rules.

Proposition 2.22. *Let $\mathcal{K}$ be an MKNF knowledge base and $\mathcal{K}'$ the MKNF knowledge base obtained from $\mathcal{K}$ by replacing every non-disjunctive rule with default negation in the head*

$$\sim\xi \leftarrow B^+, \sim B^-. \qquad \textit{with the rule} \qquad \leftarrow \xi, B^+, \sim B^-.$$

Then the MKNF models of $\mathcal{K}$ coincide with the MKNF models of $\mathcal{K}'$.

Proof. See Appendix A, page 173. □

2.5 Belief Updates

Research on belief updates has been initiated with the work of Keller and Winslett (1985); Katsuno and Mendelzon (1991) who recognised that AGM postulates do not correctly describe changes in beliefs under all circumstances. They identified two different types of change depending on whether the represented domain remains static, so the belief base only needs to be augmented with additional information about the same domain, or whether the domain itself has changed and this change needs to be recorded in its representation. The latter kind of change was dubbed *update* and described as a belief change operation that *consists of bringing a knowledge base up to date when the world described by it changes* (Katsuno and Mendelzon, 1991).

2.5.1 Katsuno and Mendelzon's Framework

From a formal viewpoint, a typical assumption when considering belief updates is that beliefs are represented in a propositional language over a finite set of propositional atoms $\mathcal{A}$. It is thus assumed that each belief base is represented by a single propositional formula. The restriction to propositional logic nullifies the distinction between standard first-order interpretations and interpretations under the standard names assumption, with both types of interpretations directly corresponding with subsets of $\mathcal{A}$. Henceforth, a *propositional interpretation* is a subset of $\mathcal{A}$ that induces a truth assignment to all propositional formulae in the standard way. We also reuse the symbol $\mathcal{I}$ to denote the set of all propositional interpretations, i.e. $\mathcal{I} = 2^{\mathcal{A}}$. The set of all models of a propositional formula ϕ is denoted by $[\![\phi]\!]$; equivalence and entailment are defined the same way as for first-order logic in Definition 2.3. In addition, it is said that a formula is *consistent* if it has at least one model; *complete* if it has exactly one model.

We liberally define a belief update operator as any function that takes two inputs, the original belief base and its update, and returns the updated belief base.

Definition 2.23 (Belief Update Operator)**.** A *belief update operator* is a binary function on the set of all propositional formulae.

Any belief update operator $\diamond$ is generalised to finite sequences of propositional formulae $\langle \phi_i \rangle_{i<n}$ as follows:

$$\Diamond\langle \phi_0 \rangle = \phi_0$$
$$\Diamond\langle \phi_i \rangle_{i<n+1} = (\Diamond\langle \phi_i \rangle_{i<n}) \diamond \phi_n \quad .$$

To further specify the desired properties of update operators, the following eight postulates for a belief update operator $\diamond$ and formulae ϕ, ψ, μ, ν were proposed in (Katsuno and Mendelzon, 1991):

(B1) $\phi \diamond \mu \models \mu$.

(B2) If $\phi \models \mu$, then $\phi \diamond \mu \equiv \phi$.

(B3) If $[\![\phi]\!] \neq \emptyset$ and $[\![\mu]\!] \neq \emptyset$, then $[\![\phi \diamond \mu]\!] \neq \emptyset$.

(B4) If $\phi \equiv \psi$ and $\mu \equiv \nu$, then $\phi \diamond \mu \equiv \psi \diamond \nu$.

(B5) $(\phi \diamond \mu) \wedge \nu \models \phi \diamond (\mu \wedge \nu)$.

(B6) If $\phi \diamond \mu \models \nu$ and $\phi \diamond \nu \models \mu$, then $\phi \diamond \mu \equiv \phi \diamond \nu$.

(B7) If ϕ is complete, then $(\phi \diamond \mu) \wedge (\phi \diamond \nu) \models \phi \diamond (\mu \vee \nu)$.

(B8) $(\phi \vee \psi) \diamond \mu \equiv (\phi \diamond \mu) \vee (\psi \diamond \mu)$.

Most of these postulates can be given a simple intuitive reading. For instance, (B1) requires that information from the update be retained in the updated belief base. This is also frequently referred to as the *principle of primacy of new information* (Dalal, 1988). Postulate is (B4) expresses that the operator must be *syntax-independent* – it must provide equivalent results given equivalent inputs. Sometimes it is useful to break it up into the following two weaker properties which, when taken together, imply (B4):

(B4.1) If $\phi \equiv \psi$, then $\phi \diamond \mu \equiv \psi \diamond \mu$.

(B4.2) If $\mu \equiv \nu$, then $\phi \diamond \mu \equiv \phi \diamond \nu$.

Similarly, (B8) can be divided into the following principles:

(B8.1) $(\phi \vee \psi) \diamond \mu \models (\phi \diamond \mu) \vee (\psi \diamond \mu)$.

(B8.2) If $\phi \models \psi$, then $\phi \diamond \mu \models \psi \diamond \mu$.

As the following proposition shows, in the presence of (B4.1) these principles are together equivalent to (B8).

Proposition 2.24. *Let $\diamond$ be a belief update operator that satisfies (B4.1). Then $\diamond$ satisfies (B8) if and only if it satisfies both (B8.1) and (B8.2).*

Proof. See Appendix A, page 183. □

The property expressed by (B8) is at the heart of belief updates: Alternative models of the original belief base ϕ are treated as possible real states of the modelled world. Each of these models is updated independently of the others to make it consistent with the update μ, obtaining a new set of interpretations – the models of the updated belief base. Based on this view of updates, Katsuno and Mendelzon proved an important representation theorem that makes it possible to constructively characterise and evaluate every operator $\diamond$ that satisfies postulates (B1) – (B8). The main idea, based on postulate (B8), is formally captured by the equation

$$[\![\phi \diamond \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \mathsf{incorporate}([\![\mu]\!], I) \enspace ,$$

where $\mathsf{incorporate}(\mathcal{M}, I)$ returns the members of $\mathcal{M}$ closer to I so that the original information in I is preserved as much as possible. A natural way of defining $\mathsf{incorporate}(\mathcal{M}, I)$ is by assigning an order $\leq^I$ over $\mathfrak{I}$ to each interpretation I and taking the minima of $\mathcal{M}$ w.r.t. $\leq^I$, i.e. $\mathsf{incorporate}(\mathcal{M}, I) = \min(\mathcal{M}, \leq^I)$. In the following we first formally establish the concept of an *order assignment*; thereafter we define when an update operator is *characterised by* such an assignment.

Given a set $\mathcal{S}$, a *preorder over $\mathcal{S}$* is a reflexive and transitive binary relation over $\mathcal{S}$; a *strict preorder over $\mathcal{S}$* is an irreflexive and transitive binary relation over $\mathcal{S}$; a *partial order over $\mathcal{S}$* is a preorder over $\mathcal{S}$ that is antisymmetric. Given a preorder $\leq$ over $\mathcal{S}$, we denote by $<$ the strict preorder induced by $\leq$, i.e. $s < t$ if and only if $s \leq t$ and not $t \leq s$. For any subset $\mathcal{T}$ of $\mathcal{S}$, the set of *minimal elements of $\mathcal{T}$ w.r.t.* $\leq$ is

$$\min(\mathcal{T}, \leq) = \{\, s \in \mathcal{T} \mid \neg \exists t \in \mathcal{T} : t < s \,\} \enspace .$$

Definition 2.25 (Order assignment)**.** Let $\mathcal{S}$ be a set. A *preorder assignment over $\mathcal{S}$* is any function ω that assigns a preorder $\leq^s_\omega$ over $\mathcal{S}$ to each $s \in \mathcal{S}$. A *partial order assignment over $\mathcal{S}$* is any preorder assignment ω over $\mathcal{S}$ such that $\leq^s_\omega$ is a partial order over $\mathcal{S}$ for every $s \in \mathcal{S}$.

Definition 2.26 (Belief Update Operator Characterised by an Order Assignment)**.** Let $\diamond$ be a belief update operator and ω a preorder assignment over $\mathfrak{I}$. We say that $\diamond$ is *characterised by ω* if for all formulae ϕ, μ,

$$[\![\phi \diamond \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \min\left([\![\mu]\!], \leq^I_\omega\right) \enspace .$$

A natural condition on the assigned orders is that every interpretation be the closest to itself. This is captured by the notion of a *faithful* order assignment:

Definition 2.27 (Faithful Order Assignment)**.** A preorder assignment ω over $\mathcal{I}$ is *faithful* if for every interpretation I the following condition is satisfied:

$$\text{For every } J \in \mathcal{I} \text{ with } J \neq I \text{ it holds that } I <^I_\omega J \enspace .$$

The representation theorem of (Katsuno and Mendelzon, 1991) states that operators characterised by faithful order assignments are exactly those that satisfy the KM postulates.

Theorem 2.28 (Katsuno and Mendelzon (1991))**.** *Let $\diamond$ be a belief update operator. Then the following conditions are equivalent:*

a) The operator $\diamond$ satisfies conditions (B1) – (B8).

b) The operator $\diamond$ is characterised by a faithful preorder assignment.

c) The operator $\diamond$ is characterised by a faithful partial order assignment.

2.5.2 Specific Belief Update Operators

Katsuno and Mendelzon's results provide a framework for belief update operators, each specified on the semantic level by a faithful partial order assignment over $\mathcal{I}$. The most influential instance of this framework is the *Possible Models Approach* introduced by Winslett (1988), based on minimising the set of atoms whose truth value changes when an interpretation is updated. It is defined as follows:

Definition 2.29 (Winslett's Operator (Winslett, 1988))**.** Let the partial order assignment W be defined for all interpretations I, J, K as

$$J \leq^I_{\mathsf{W}} K \qquad \text{if and only if} \qquad (J \div I) \subseteq (K \div I) \enspace ,$$

where $\div$ denotes set-theoretic symmetric difference. *Winslett's update operator* is an arbitrary but fixed belief update operator $\diamond_{\mathsf{W}}$ that is characterised by W.

By unfolding the definitions above, we obtain the following characterisation of $\diamond_{\mathsf{W}}$:

$$[\![\phi \diamond_{\mathsf{W}} \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \{\, J \in [\![\mu]\!] \mid \neg\exists K \in [\![\mu]\!] : (K \div I) \subsetneq (J \div I) \,\} \enspace .$$

It is not difficult to verify that W is a faithful partial order assignment, so it follows from Theorem 2.28 that $\diamond_{\mathsf{W}}$ satisfies postulates (B1) – (B8). Note that there is a whole class of operators characterised by $\leq^I_{\mathsf{W}}$ that differ in the syntactic representation of updated belief bases. Insofar as we are interested in semantic properties of Winslett's operator, it follows from (B4) that it does not matter which operator from this class we pick as $\diamond_{\mathsf{W}}$.

A related instance of Katsuno and Mendelzon's framework is Forbus' update operator which minimises the *number of* atoms whose truth values are modified (Forbus, 1989). Formally:

Definition 2.30 (Forbus' Operator (Forbus, 1989))**.** Let the preorder assignment F be defined for all interpretations I, J, K as

$$J \leq^I_{\mathsf{F}} K \qquad \text{if and only if} \qquad |J \div I| \leq |K \div I| \enspace ,$$

where $|\cdot|$ denotes the cardinality of a set. *Forbus' update operator* is an arbitrary but fixed belief update operator $\diamond_{\mathsf{F}}$ that is characterised by F.

By unfolding the definitions above, we obtain the following characterisation of $\diamond_{\mathsf{F}}$:

$$[\![\phi \diamond_{\mathsf{F}} \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \{ J \in [\![\mu]\!] \mid \neg\exists K \in [\![\mu]\!] : |K \div I| < |J \div I| \} \enspace .$$

Similarly as with Winslett's operator, since F is a faithful preorder assignment, it follows from Theorem 2.28 that $\diamond_{\mathsf{F}}$ satisfies postulates (B1) – (B8). It is also not difficult to verify that for all formulae ϕ, μ,

$$\phi \diamond_{\mathsf{F}} \mu \models \phi \diamond_{\mathsf{W}} \mu \enspace . \tag{2.1}$$

The converse does not generally hold, as witnessed by $\phi = p \wedge q \wedge r$ and $\mu = \neg p \vee (\neg q \wedge \neg r)$ because

$$\phi \diamond_{\mathsf{F}} \mu \equiv \neg p \wedge q \wedge r \qquad \text{while} \qquad \phi \diamond_{\mathsf{W}} \mu \equiv (\neg p \wedge q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \enspace .$$

This means that in some cases Forbus' semantics preserves strictly more information from the original belief base ϕ than Winslett's operator. Depending on the particular application domain, one or the other operator may be more appropriate (Winslett, 1990).

Despite its significance and intuitive appeal, Winslett's operator has also received a considerable amount of criticism. Firstly, it treats disjunction in updates exclusively and not inclusively. For instance, if $\phi = \neg p \wedge \neg q$ and $\mu = p \vee q$, then

$$\phi \diamond_{\mathsf{W}} \mu \equiv (p \vee q) \wedge (\neg p \vee \neg q) \enspace ,$$

which is considered too restrictive in certain scenarios where the expected result is simply $p \vee q$ (Herzig, 1996; Zhang and Foo, 1996). It follows from (2.1) that such criticism also applies to Forbus' semantics and, as shown by Herzig and Rifi (1999), it is actually a consequence of the KM postulates, namely of postulate (B5). In other words, every update semantics that addresses this problem, such as MCD, MCD*, MCE (Zhang and Foo, 2000) or WSS and its syntax-independent modification (Winslett, 1990; Herzig, 1996), must violate (B5).

The general desirability of other postulates has also been questioned. For example, (B2) is considered too strong by many (Brewka and Hertzberg, 1993; Boutilier, 1995; Doherty et al., 1998; Herzig and Rifi, 1999), rendering it controversial. Furthermore, (B2) is strongly related to the following principles (Herzig and Rifi, 1999):[6]

(B2.⊤) $\phi \diamond \top \equiv \phi$.

(B2.1) $\phi \wedge \mu \models \phi \diamond \mu$.

(B2.2) $(\phi \wedge \mu) \diamond \mu \models \phi$.

The first two are uncontroversial as they are satisfied by all considered update operators. In addition, in the presence of (B4.1), the latter two together are powerful enough to entail (B2). Hence the controversial part of (B2) is (B2.2). Furthermore, (B6) entails (B2) in the presence of (B2.⊤), so the criticism of (B2.2) transfers to (B6). Lastly, (B7) is considered almost meaningless (Herzig and Rifi, 1999) because it is restricted to the marginal case of complete belief bases.

[6]From the analysis by Herzig and Rifi (1999) it might seem that (B2) entails all of these principles. Although this is the case with (B2.⊤) and (B2.2), (B2.1) cannot be derived from (B2) alone. For instance, a (rather ridiculous) operator $\diamond$ such that $p \diamond q \equiv \bot$ may still satisfy (B2) but does not satisfy (B2.1).

Nevertheless, it is not difficult to verify that *(B2.1) does follow from (B2) in the presence of (B8.2)* as follows: Since $\phi \wedge \mu \models \mu$, from (B2) it follows that $(\phi \wedge \mu) \diamond \mu \equiv \phi \wedge \mu$ and by (B8.2) since $\phi \wedge \mu \models \phi$, it follows that $(\phi \wedge \mu) \diamond \mu \models \phi \diamond \mu$. By transitivity of entailment we obtain (B2.1).

Another issue with belief update operators is connected to the handling of integrity constraints: suppose that ψ represents integrity constraints that must be adhered to both before and after an update. The usual proposal to define such an operation by $\phi \diamond_{\psi} \mu = \phi \diamond (\mu \wedge \psi)$ (Winslett, 1990) has its flaws, as demonstrated by (Lifschitz, 1990; Herzig, 1996). Many methods for addressing these issues are based on providing extra-logical information to the update operator, such as dependencies between atoms (Herzig, 1996) or causal rules (Doherty et al., 1998). The recent developments seem to indicate that this is still an open problem due to the issues with integrating causal reasoning with belief updates (Herzig, 2005; Veltman, 2005).

2.5.3 Formula-Based Operators

The belief update operators discussed until now are also called *model-based* because they are characterised on the semantic level. Their characteristic property is that they are syntax-independent, i.e. they satisfy the postulate (B4). Earlier approaches to updates, dubbed *formula-based* (Winslett, 1990), operate on the syntax of a belief base and, as a consequence, are not syntax-independent. Nevertheless, recently they were considered for performing ontology updates, in particular for updating TBoxes (Calvanese et al., 2010).

To define these operators, we need to generalise our definition of a belief base to a *finite set of propositional formulae*. A (formula-based) update operator is defined as a function that assigns a belief base to every pair of belief bases.

Traditional formula-based update operators are Set-Of-Theories (Fagin et al., 1983), WIDTIO (Ginsberg, 1986; Ginsberg and Smith, 1988; Winslett, 1990) and Cross-Product (Ginsberg, 1986). We define only the latter two because the Set-Of-Theories operator produces a *collection of belief bases* as its result instead of a single belief base, and is equivalent to the Cross-Product operator which compiles these belief bases into one. These operators are based on the following idea: A natural candidate for the result of updating $\mathcal{B}$ by $\mathcal{U}$ is a union of $\mathcal{U}$ with a maximal subset $\mathcal{B}'$ of $\mathcal{B}$ that is consistent with $\mathcal{U}$. These remainders of $\mathcal{B}$ are formalised as follows:

Definition 2.31 (Possible Remainder)**.** Let $\mathcal{B}$ and $\mathcal{U}$ be belief bases. We say that $\mathcal{B}'$ is a *possible remainder of* $\mathcal{B}$ *w.r.t.* $\mathcal{U}$ if the following conditions are satisfied:

1. $\mathcal{B}' \subseteq \mathcal{B}$;
2. $\mathcal{B}' \cup \mathcal{U}$ is consistent;
3. $\mathcal{B}'$ is subset-maximal among all sets satisfying 1. and 2.

We denote the set of all possible remainders of $\mathcal{B}$ w.r.t. $\mathcal{U}$ by $\mathsf{rem}(\mathcal{B}, \mathcal{U})$.

The distinct formula-based operators only differ in how they deal with the case when there is more than one candidate for $\mathcal{B}'$. The operator WIDTIO (When In Doubt, Throw It Out (Winslett, 1990)) takes the safe path – it keeps exactly those formulae that belong to the intersection of all remainders and throws away the rest.

Definition 2.32 (WIDTIO Operator)**.** The formula-based operator $\circ_{\mathsf{WIDTIO}}$ is defined for all belief bases $\mathcal{B}, \mathcal{U}$ as

$$\mathcal{B} \circ_{\mathsf{WIDTIO}} \mathcal{U} = \mathcal{U} \cup \bigcap \mathsf{rem}(\mathcal{B}, \mathcal{U}) \enspace .$$

The Cross-Product operator instead combines information from different remainders in the resulting belief base.

Definition 2.33 (Cross-Product Operator)**.** The formula-based operator $\circ_{\mathsf{CP}}$ is defined for all belief bases $\mathcal{B}, \mathcal{U}$ as

$$\mathcal{B} \circ_{\mathsf{CP}} \mathcal{U} = \mathcal{U} \cup \{ \psi \}$$

where ψ is the formula

$$\bigvee_{\mathcal{B}' \in \mathsf{rem}(\mathcal{B}, \mathcal{U})} \bigwedge_{\phi \in \mathcal{B}'} \phi \enspace .$$

It should be noted that although WIDTIO and Cross-Product are traditionally called *update* operators, from the perspective of belief change they carry more characteristics of *revision* than of update. WIDTIO actually coincides with the *internal full meet base revision operator*, i.e. the base revision operator derived through the Levi identity from the partial meet base contraction operator with the selection function $\gamma(\mathcal{K}) = \mathcal{K}$ (c.f. (Hansson, 1993a)). Cross-Product can be seen as a change operator that lies somewhere between belief base revision and belief set revision (Gärdenfors, 1992) – when the remainder is unique, it adds or removes whole formulae from the belief base, just as a base revision operator would, but when multiple remainders exist, consequences common to all remainders are always kept, similarly as with the original partial meet revision operators (Alchourrón et al., 1985).

2.6 Updates of First-Order Theories

In order to perform updates of ontologies using belief update operators, we need to generalise them to deal with updates of first-order knowledge bases.

But before we start introducing the formal definitions, let us briefly discuss the main challenges that lie ahead. Suppose that we want to generalise the definition of Winslett's operator $\diamond_{\mathsf{W}}$ to update first-order theories. Recall first that in the propositional case, $\diamond_{\mathsf{W}}$ is semantically characterised by

$$[\![\phi \diamond_{\mathsf{W}} \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \{ J \in [\![\mu]\!] \mid \neg \exists K \in [\![\mu]\!] : (K \div I) \subsetneq (J \div I) \} \enspace .$$

To adapt this definition to first-order logic, we somehow need to define "symmetric difference" between first-order interpretations. We can do this on a predicate by predicate basis, turning $(K \div I) \subsetneq (J \div I)$ into something like

$$\forall P \in \mathcal{P} : \left(P^K \div P^I \right) \subsetneq \left(P^J \div P^I \right) \enspace .$$

However, there is one subtle problem here: Comparing interpretations of a predicate symbol P under I, J and K only makes sense when constant symbols are interpreted the same way under all of these interpretations. This has been recognised already in (Winslett, 1990) and addressed by imposing that I, J and K must be over the same universe Δ and must interpret constant symbols identically. This solution is also adopted in the work on ABox updates in DLs at least as expressive as $\mathcal{ALC}$ (Liu et al., 2006; Bong, 2007; Drescher et al., 2009). Its downside is that one essentially loses support for updates of equality assertions, as demonstrated by the following example.

Example 2.34 (Updates of Equality Assertions)**.** Suppose that the knowledge base $\phi = (a \approx b)$ is updated by $\mu = (a \not\approx b)$. Then every model I of ϕ interprets constants a and b as the same object d. Furthermore, if an update of I can only result in interpretations J that interpret constants exactly as I does, then it follows that no such J is a model of μ. As a consequence, $\phi \diamond_{\mathsf{W}} \mu$ must be unsatisfiable.

Another issue, albeit a less severe one, is that with this restriction in place, we cannot define Winslett's first-order operator using a partial order assignment: there is no way to impose that $\min(\llbracket \mu \rrbracket, \leq^I_{\mathsf{W}})$ must be empty when $\llbracket \mu \rrbracket$ is non-empty, as required in the above example. This also means that the postulate (B3) would no longer be satisfied by the operator.

A different way to address the problem is to assume that we use the standard names assumption (De Giacomo et al., 2006, 2007, 2009). If, in addition, we interpret the equality predicate as a congruence relation instead of true equality, as a bonus we regain support for updates of equality assertions. In the definition of Winslett's first-order operator we thus assume that we work with the set of interpretations $\mathcal{I}$ defined in Section 2.1, i.e. interpretations under the standard names assumption and with $\approx$ interpreted as a congruence relation. As demonstrated by Proposition 2.17, this modification is faithful to the original semantics for DLs. Furthermore, we regain the ability to define Winslett's operator using a partial order assignment.

Although this approach technically allows for performing equality updates, these may have unexpected side-effects, such as the following one:

Example 2.35. Suppose that we update the theory $\mathcal{T} = \{ a \approx b \}$ by the theory $\mathcal{U} = \{ a \not\approx b \}$ using Winslett's operator and assume that our language contains a unary predicate P. Since equality allows for replacement of equals by equals, we know that

$$\mathcal{T} \models P(a) \equiv P(b) \ .$$

Furthermore, although $\mathcal{T} \diamond_{\mathsf{W}} \mathcal{U}$ no longer entails $a \approx b$, it still holds that

$$\mathcal{T} \diamond_{\mathsf{W}} \mathcal{U} \models Pa \equiv P(b) \ .$$

In other words, the side-effects of equality from $\mathcal{T}$ carry over to $\mathcal{T} \diamond_{\mathsf{W}} \mathcal{U}$. This sort of behaviour may be unexpected by the user of the formalism and brings with it also technical difficulties due to the fact that $\mathcal{T} \diamond_{\mathsf{W}} \mathcal{U}$ cannot be syntactically represented without using the predicate P even though this predicate occurs neither in $\mathcal{T}$ nor in $\mathcal{U}$.

2.6.1 First-Order Update Operators

We proceed with the general definition of a first-order update operator. We introduce it for the general case of first-order theories, instead of first-order sentences, since this will be useful in Chapter 3.

Definition 2.36 (First-Order Update Operator)**.** A first-order update operator is a binary function on the set of all first-order theories.

Any first-order update operator $\diamond$ is generalised to finite sequences of first-order theories $\langle \mathcal{T}_i \rangle_{i<n}$ as follows:

$$\Diamond \langle \mathcal{T}_0 \rangle = \mathcal{T}_0 \ ,$$
$$\Diamond \langle \mathcal{T}_i \rangle_{i<n+1} = (\Diamond \langle \mathcal{T}_i \rangle_{i<n}) \diamond \mathcal{T}_n \ .$$

A single first-order sentence ϕ is updated by treating it as the theory $\{ \phi \}$.

Similarly as in the propositional case, first-order update operators can be defined by specifying an order assignment on the set of all interpretations $\mathcal{I}$.

Definition 2.37 (First-Order Update Operator Characterised by an Order Assignment)**.** Let $\diamond$ be a first-order update operator and ω a preorder assignment over $\mathcal{I}$. We say that $\diamond$

is *characterised by* ω if for all first-order theories $\mathcal{T}, \mathcal{U}$,

$$[\![\mathcal{T} \diamond \mathcal{U}]\!] = \bigcup_{I \in [\![\mathcal{T}]\!]} \min\left([\![\mathcal{U}]\!], \leq^I_\omega\right) \quad .$$

In particular, Winslett's first-order operator is defined as follows:[7]

Definition 2.38 (Winslett's First-Order Operator (Winslett, 1990))**.** Let the partial order assignment W be defined for all interpretations $I, J, K \in \mathcal{I}$ as

$$J \leq^I_{\mathsf{W}} K \qquad \text{if and only if} \qquad \forall P \in \mathcal{P}^{\approx} : \left(P^J \div P^I\right) \subseteq \left(P^K \div P^I\right) \quad .$$

Winslett's first-order operator is an arbitrary but fixed first-order update operator $\diamond_{\mathsf{W}}$ that is characterised by W.

Other model-based belief update operators could also theoretically be generalised to the first-order case though this has not been considered in the literature on ontology updates. One of the reasons might be that relying on the comparison of set cardinality when the sets in question are infinite may lead to awkward results. For instance, if the set $P^J \div P^I$ is countably infinite, based on cardinality alone the interpretation J cannot be preferred another interpretation K that interprets all predicate symbols except for P just like J and $P^K \div P^I$ is a countably infinite *strict superset of* $P^J \div P^I$.

2.6.2 Syntax-Based Properties of First-Order Updates

In this and the following section we look at some properties that can be naturally expected of first-order update operators, and in particular properties that are satisfied by Winslett's operator. The properties in this section have a syntactic basis and represent basic expectations regarding the behaviour of a domain-independent update operator, i.e. a general-purpose update operator that does not take domain knowledge into account and treats all predicates uniformly.

The first property that we consider is *language conservation*. Informally, if both the initial and updating theories represent knowledge about predicate symbols from the set A, then the updated theory should not introduce knowledge about predicate symbols that do not belong to A. The formalisation of this property relies on the notion of *interpretation restriction* and the related concept of *saturated set of interpretations*. Intuitively, a set of interpretations $\mathcal{M} \in \mathfrak{M}$ is *saturated relative to* A if it contains knowledge about predicate symbols from A only.

Definition 2.39 (Interpretation Restriction)**.** Let A be a set of predicate symbols, $I \in \mathcal{I}$ and $\mathcal{M} \in \mathfrak{M}$. The *restriction of* I *to* A is an interpretation $I^{[A]}$ such that for every ground atom p,

$$I^{[A]} \models p \qquad \text{if and only if} \qquad \mathsf{pr}(p) \subseteq A \land I \models p \quad .$$

The *restriction of* $\mathcal{M}$ *to* A is defined as $\mathcal{M}^{[A]} = \left\{ I^{[A]} \mid I \in \mathcal{M} \right\}$.

Definition 2.40 (Saturated MKNF Interpretation)**.** Let A be a set of predicate symbols and $\mathcal{M} \in \mathfrak{M}$. We say that $\mathcal{M}$ is *saturated relative to* A if for every interpretation $I \in \mathcal{I}$,

$$I^{[A]} \in \mathcal{M}^{[A]} \qquad \text{implies} \qquad I \in \mathcal{M} \quad .$$

[7] Note that we exercise a slight abuse of notation by using the same symbols W and $\diamond_{\mathsf{W}}$ for Winslett's order assignment and update operator in both the propositional and first-order cases.

To formally illustrate this concept, we show that the set of models of a first-order theory $\mathcal{T}$ is saturated relative to the set of predicate symbols that are relevant to $\mathcal{T}$.

Proposition 2.41. *Let A be a set of predicate symbols and $\mathcal{T}$ a first-order theory such that $\mathsf{pr}(\mathcal{T}) \subseteq A$. Then $[\![\mathcal{T}]\!]$ is saturated relative to A.*

Proof. See Appendix A, page 176. □

In other words, the set of models of a single theory conserves its language. Language conservation for an update operator $\diamond$ extends the same property to sequences of first-order theories. In particular, it expresses that $\diamond$ preserves the saturation relative to A whenever A contains all predicate symbols relevant to either the initial theory or in one of its updates.

Definition 2.42 (Language Conservation)**.** Let $\diamond$ be a first-order update operator. We say that $\diamond$ *conserves the language* if for all sets of predicate symbols A and every sequence of first-order theories $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$ such that for all $i < n$, $\mathsf{pr}(\mathcal{T}_i) \subseteq A$, $[\![\diamond \boldsymbol{T}]\!]$ is saturated relative to A.

Since Winslett's operator treats all predicate symbols in a uniform manner, in the absence of the equality predicate it naturally satisfies this property. With equality, however, problematic cases such as the one shown in Example 2.35 arise. Due to such issues, we will not allow for the use of equality in Chapter 4 where we need to rely on language conservation.

Theorem 2.43. *If we do not allow for the equality predicate $\approx$, then Winslett's first-order update operator $\diamond_{\mathsf{W}}$ conserves the language.*

Proof. See Appendix A, page 184. □

Another important property of Winslett's operator is that it satisfies the basic intuitions regarding *updates of relational databases*. In particular, when it is used for updating consistent sets of literals, the literals keep their truth values by inertia. As we shall see in Section 2.8, this constitutes a case that is handled the same way by rule update semantics and so provides a basic layer of interoperability between belief update operators and rule update semantics. We formally formulate this property as follows:

Definition 2.44 (Fact Update)**.** Let $\diamond$ be a first-order update operator. We say that $\diamond$ *respects fact update* if for every finite sequence of consistent sets of ground objective literals $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$,

$$[\![\diamond \boldsymbol{T}]\!] = \left\{ I \in \mathfrak{I} \;\middle|\; I \models \left\{ l \in \mathcal{L}_{\mathsf{G}} \;\middle|\; \exists j < n : l \in \mathcal{T}_j \wedge (\forall i : j < i < n \Longrightarrow \bar{l} \notin \mathcal{T}_i) \right\} \right\} \enspace .$$

Theorem 2.45. *Winslett's first-order update operator $\diamond_{\mathsf{W}}$ respects fact update.*

Proof. See Appendix A, page 184. □

2.6.3 Generalising Katsuno and Mendelzon's Postulates

Most of Katsuno and Mendelzon's belief update postulates can be directly generalised to the case of first-order theories and update operators. Issues arise only with the postulates (B7) and (B8) because they require disjunction of a pair of theories to be defined. Although semantically this is not problematic – the models of a disjunction of two theories should be the models of either of the theories – on the syntactic level it is unclear how $\mathcal{T} \vee \mathcal{S}$ should be specified.

In case of (B8) this situation can be partially resolved by reformulating the principle (B8.2) which does not require disjunction and follows from (B8) together with (B4.1) (c.f. Proposition 2.24). We can thus define the following update postulates for first-order update operators:

(FO1) $\mathcal{T} \diamond \mathcal{U} \models \mathcal{U}$.

(FO2) If $\mathcal{T} \models \mathcal{U}$, then $\mathcal{T} \diamond \mathcal{U} \equiv \mathcal{T}$.

(FO2.⊤) $\mathcal{T} \diamond \emptyset \equiv \mathcal{T}$.

(FO2.1) $\mathcal{T} \cup \mathcal{U} \models \mathcal{T} \diamond \mathcal{U}$.

(FO2.2) $(\mathcal{T} \cup \mathcal{U}) \diamond \mathcal{U} \models \mathcal{T}$.

(FO3) If $[\![\mathcal{T}]\!] \neq \emptyset$ and $[\![\mathcal{U}]\!] \neq \emptyset$, then $[\![\mathcal{T} \diamond \mathcal{U}]\!] \neq \emptyset$.

(FO4) If $\mathcal{T} \equiv \mathcal{S}$ and $\mathcal{U} \equiv \mathcal{V}$, then $\mathcal{T} \diamond \mathcal{U} \equiv \mathcal{S} \diamond \mathcal{V}$.

(FO4.1) If $\mathcal{T} \equiv \mathcal{S}$, then $\mathcal{T} \diamond \mathcal{U} \equiv \mathcal{S} \diamond \mathcal{U}$.

(FO4.2) If $\mathcal{U} \equiv \mathcal{V}$, then $\mathcal{T} \diamond \mathcal{U} \equiv \mathcal{T} \diamond \mathcal{V}$.

(FO5) $(\mathcal{T} \diamond \mathcal{U}) \cup \mathcal{V} \models \mathcal{T} \diamond (\mathcal{U} \cup \mathcal{V})$.

(FO6) If $\mathcal{T} \diamond \mathcal{U} \models \mathcal{V}$ and $\mathcal{T} \diamond \mathcal{V} \models \mathcal{U}$, then $\mathcal{T} \diamond \mathcal{U} \equiv \mathcal{T} \diamond \mathcal{V}$.

(FO8.2) If $\mathcal{T} \models \mathcal{S}$, then $\mathcal{T} \diamond \mathcal{U} \models \mathcal{S} \diamond \mathcal{U}$.

Note that these postulates are defined in terms of the entailment on models from $\mathfrak{I}$, i.e. interpretations under the standard names assumption with equality interpreted as a congruence relation. If we defined them in terms of the standard first-order entailment $\models_{\text{FO}}$, it follows from Proposition 2.5 that we would end up with a different set of postulates when considering updates of infinite first-order theories.

It is not difficult to show that every first-order update operator characterised by a faithful order assignment satisfies all of these postulates.

Proposition 2.46. *Every first-order update operator that is characterised by a faithful preorder assignment satisfies postulates (FO1) – (FO6) and (FO8.2).*

Proof. See Appendix A, page 185. □

As a consequence, Winslett's first-order operator also satisfies the proposed postulates.

Corollary 2.47. *Winslett's first-order operator $\diamond_{\mathsf{W}}$ satisfies postulates (FO1) – (FO6) and (FO8.2).*

Proof. Follows from Proposition 2.46 and from the fact that W is a faithful partial order assignment. □

Note that the other half of Katsuno and Mendelzon's representation theorem cannot be immediately generalised to first-order logic since we are still missing a counterpart of the crucial postulate (B8.1). The presented postulates can nevertheless serve to examine different classes of first-order update operators. For instance, in Chapter 3 we prove results that hold for all first-order update operators that satisfy property (FO8.2).

2.7 Ontology Updates

Formal methods for ontology updates are based on the classical update operators described previously. The main issue with adapting first-order update operators to deal with ontologies is that description logics are only *fragments* of first-order logic, so the result of a first-order update operator may not be *expressible* in the DL used to encode the original ontology and its update. Other problems stem from examples showing that when model-based operators are used to update TBoxes, unexpected results are obtained. Due to this, formula-based operators have been used as an alternative way of dealing with ontology updates. We briefly describe both lines of work in the following sections.

2.7.1 Expressibility and ABox Updates

Now that we have a definition first-order update operators, and particularly we defined Winslett's first-order operator $\diamond_{\mathrm{W}}$, we can think about how to use such operators to update *DL ontologies* instead of first-order theories.

The basic idea is very simple: assuming that $\diamond$ is some first-order update operator, we define an update of an ontology $\mathcal{O}_1$ by an ontology $\mathcal{O}_2$ as

$$\mathcal{O}_1 \diamond \mathcal{O}_2 = \kappa(\mathcal{O}_1) \diamond \kappa(\mathcal{O}_2) \enspace .$$

In other words, we translate both the original ontology and its update to their first-order representations and update those instead. The problem here is that the result of $\kappa(\mathcal{O}_1) \diamond \kappa(\mathcal{O}_2)$ may not be expressible in the DL that was used to encode the original ontology $\mathcal{O}_1$ and its update $\mathcal{O}_2$. This was observed already by Baader et al. (2005a) in case of the Description Logic $\mathcal{ALCQI}$: it turned out that querying an ontology updated using $\diamond_{\mathrm{W}}$ is undecidable and, consequently, the result of the update cannot be expressed in $\mathcal{ALCQI}$.

Further work on updates in DLs at least as expressive as $\mathcal{ALC}$ has thus focused on updates of ABoxes only, in particular on simple ABox updates that only allow for updates with assertions about concept names (Liu et al., 2006; Bong, 2007; Drescher et al., 2009). The update semantics used to perform the updates coincides with Winslett's first-order operator $\diamond_{\mathrm{W}}$ and several DLs have been identified in which expressibility of the updated ABox is guaranteed. In some of these DLs, the updated ABox may be exponential in size of the original ABox as well as of the update, while in others it may be exponential (only) in the size of the update.

A related approach concentrates on updates of ABoxes in a family of lightweight Description Logics called DL-Lite (De Giacomo et al., 2006, 2007, 2009). These even allow for a static TBox $\mathcal{T}$, treating it the same way that integrity constraints were handled in early belief update semantics (Katsuno and Mendelzon, 1991) (c.f. the end of Section 2.5.2):

$$\mathcal{A}_1 \diamond_{\mathrm{W}}^{\mathcal{T}} \mathcal{A}_2 = \mathcal{A}_1 \diamond_{\mathrm{W}} (\mathcal{A}_2 \cup \mathcal{T}) \enspace .$$

On the positive side, this means that the uncomplicated operator $\diamond_{\mathrm{W}}$ suffices to deal with the presence of a static TBox; on the negative side, such an approach is most probably going to suffer from the same types of issues that arise from using the same method to augment belief update semantics with integrity constraints (Lifschitz, 1990; Herzig, 1996, 2005). This line of work also provides polynomial algorithms for computing the updated ABoxes or their approximations, depending on which flavour of DL-Lite is considered.

2.7.2 Ontology Updates Using Formula-Based Operators

More recently, attention has also moved towards TBox updates. Calvanese et al. (2010) argued that model-based update operators provide inappropriate results when applied to TBoxes (c.f. Example 1.8) and suggested to use a formula-based operator instead. Unlike with model-based operators, there are no semantic issues with lifting formula-based operators such as WIDTIO to deal with first-order logic or ontologies and expressibility is also not an issue. Calvanese et al. (2010) have particularly introduced the following formula-based operator, called *Bold*, for performing TBox updates in DL-Lite:

Definition 2.48 (Bold Operator (Calvanese et al., 2010))**.** A *remainder selection function* is a function s that assigns to every set of remainders $\mathcal{R}$ a remainder $s(\mathcal{R}) \in \mathcal{R}$.

Given a remainder selection function s, the formula-based operator $\circ^s_{\text{BOLD}}$ is for all knowledge bases $\mathcal{B}, \mathcal{U}$ defined as

$$\mathcal{B} \circ^s_{\text{BOLD}} \mathcal{U} = \mathcal{U} \cup s(\mathsf{rem}(\mathcal{B}, \mathcal{U})) \enspace .$$

Similarly as WIDTIO, the Bold operator is strongly related to *base revision operators* since it coincides with the *internal base revision operator* associated with the *partial meet base contraction* with the selection function $\gamma(\mathcal{K}) = \{ s(\mathcal{K}) \}$ (c.f. (Hansson, 1993a)).

More recently, Lenzerini and Savo (2011) have used an operator inspired by WIDTIO to tackle ABox updates, too. Nevertheless, their operator performs a deductive closure of the ABox before it computes the remainders, so it seems to be strongly related to the standard *full meet AGM revision operator* (Alchourrón et al., 1985).

2.8 Rule Updates

State-of-the-art rule update semantics are based on fundamentally different principles and methods when compared to their belief update counterparts. As illustrated in Example 1.6, the main reason for this is that modifications on the level of individual stable models (Alferes and Pereira, 1996), akin to model-based belief update operators, are unable to capture the essential relationships between literals encoded in rules (Leite and Pereira, 1997).

There exist a number of approaches for rectifying this issue, each with a significantly different technical realisation. This section provides an overview of existing rule update semantics, pointing at some of the technical as well as semantic differences between them.

However, our main goal is to identify properties that these semantics have *in common* because these can serve as general guiding principles for updates of hybrid knowledge bases. Thus, instead of performing a thorough formal analysis and comparison of rule update semantics, we rely on examples to show how these semantics are interrelated and refrain from presenting all technical details which would only detain us from reaching our goals. We do pay more attention to semantics based on *causal rejection* that have been thoroughly studied in the literature and are subject of Chapter 8 where we develop a semantic characterisation of one of them.

Rule update semantics typically deal only with ground non-disjunctive rules and some do not allow for default negation in their heads. While some of them follow the belief update tradition and construct an updated program given the original program and its update, others only assign a set of stable models to a pair or sequence of programs where each component represents an update of the preceding ones. In order to compare these semantics, we adopt the latter, less restrictive point of view. The "input" of a rule update semantics is thus defined as follows:

Definition 2.49 (Dynamic Logic Program)**.** A *dynamic logic program* (DLP) is a finite sequence of ground non-disjunctive logic programs.

Given a DLP $\boldsymbol{P}$, we denote by $\mathsf{all}(\boldsymbol{P})$ the set of all rules belonging to the programs in $\boldsymbol{P}$. We say that $\boldsymbol{P}$ is *acyclic* if $\mathsf{all}(\boldsymbol{P})$ is acyclic.

In order to avoid issues with rules that are repeated in multiple components of a DLP, we assume throughout this section that every rule is uniquely identified in all set-theoretic operations. This could be formalised by assigning a unique name to each rule and performing operations on names instead of the rules themselves. The technical realisation is left to the reader.

The set of stable models assigned to dynamic logic programs under a particular update semantics will be denoted as follows:

Definition 2.50 (Rule Update Semantics)**.** A *rule update semantics* S is characterised by a (partial) function $[\![\cdot]\!]_{\mathsf{S}}$ that assigns a set of ASP interpretations $[\![\boldsymbol{P}]\!]_{\mathsf{S}}$ to a dynamic logic program $\boldsymbol{P}$. We call each member of $[\![\boldsymbol{P}]\!]_{\mathsf{S}}$ an S*-model of* $\boldsymbol{P}$.

We first discuss a major group of semantics based on the *causal rejection principle* (Leite and Pereira, 1997; Buccafurri et al., 1999; Alferes et al., 2000; Eiter et al., 2002; Alferes et al., 2005; Osorio and Cuevas, 2007) in Section 2.8.1, followed by semantics based on *preferences* (Zhang, 2006; Delgrande et al., 2007) in Section 2.8.2. Section 2.8.3 is devoted to semantics that bear characteristics of revision rather than update (Sakama and Inoue, 2003; Osorio and Zepeda, 2007; Delgrande, 2010) and touches upon approaches that manipulate dependencies on default assumptions induced by rules (Šefránek, 2011; Krümpelmann and Kern-Isberner, 2010; Krümpelmann, 2012). Finally, in Section 2.8.4 we formulate fundamental common properties of the introduced rule update semantics.

2.8.1 Causal Rejection-Based Semantics

The *causal rejection principle* (Leite and Pereira, 1997) forms the basis of a number of rule update semantics. Informally it can be stated as follows:

> A rule should be *rejected* when it is directly contradicted by a more recent rule.

Formalisms based on this principle focus only on conflicts between heads of rules. Initially, only conflicts between *objective* literals in rule heads were considered and default negation in rule heads was not allowed (Leite and Pereira, 1997; Eiter et al., 2002). Later it was found that this approach has certain limitations, namely that some belief states, represented by stable models, become unreachable (Alferes et al., 2000; Leite, 2003). For example, no update of the program $\mathcal{P} = \{ p. \}$ leads to a stable model where neither p nor $\neg p$ is true. Default negation in rule heads was thus used to regain reachability of such states. For instance, the update $\mathcal{U} = \{ \sim p., \sim\neg p. \}$ forces p to be unknown, regardless of its previous state. In the following we thus present the semantics from (Leite and Pereira, 1997; Eiter et al., 2002) in their generalised forms that coincide with their original definitions on programs without default negation in rule heads (Leite, 2003).

A conflict between rules occurs when the head literal of one rule is the default or strong negation of the head literal of the other rule. Similarly as in (Leite, 2003), we consider the conflicts between an objective literal and its default negation as primary while conflicts between objective literals are handled by *expanding* the DLP accordingly. In particular, whenever the DLP contains a rule with an objective literal l in its head, its expansion also contains a rule with the same body and the literal $\sim\bar{l}$ in its head, where $\bar{l}$ denotes the literal complementary to l, i.e. $\bar{l} = \neg p$ if l is the atom p and $\bar{l} = p$ if l is the objective literal $\neg p$. Formally:

Definition 2.51 (Expanded Version of a DLP)**.** Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP. The *expanded version of* $\boldsymbol{P}$ is the DLP $\boldsymbol{P}^{\mathsf{e}} = \langle \mathcal{P}_i^{\mathsf{e}} \rangle_{i<n}$ where for every $i < n$,

$$\mathcal{P}_i^{\mathsf{e}} = \mathcal{P}_i \cup \left\{ \sim\bar{l} \leftarrow B(\pi). \;\middle|\; \pi \in \mathcal{P}_i \wedge H(\pi) = \{ l \} \wedge l \in \mathcal{L}_{\mathsf{G}} \right\} \quad .$$

The additional rules in the expanded version capture the coherence principle: when an objective literal l is derived, its complement $\bar{l}$ cannot be concurrently true and thus $\sim\bar{l}$ must be true. In this way, every conflict between complementary objective literals directly translates into a conflict between an objective literal and its default negation. This enables us to define a conflict between a pair of rules as follows: we say that rules π, σ *are in conflict*, denoted by $\pi \bowtie \sigma$, if and only if

$$H(\pi) = \sim H(\sigma) \neq \emptyset \ .$$

The historically first rule update semantics is the *justified update semantics*, or *JU-semantics* for short (Leite and Pereira, 1997), with the idea to define a set of *rejected rules*, which depends on a stable model candidate, and then verify that the candidate is indeed a stable model of the remaining rules.

Definition 2.52 (JU-Semantics (Leite and Pereira, 1997))**.** Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP and J an ASP interpretation. We define the set of rejected rules $\mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J)$ as

$$\mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J) = \{ \, \pi \in \mathcal{P}_i \mid \exists j \, \exists \sigma : i < j < n \wedge \sigma \in \mathcal{P}_j \wedge \pi \bowtie \sigma \wedge J \models B(\sigma) \, \} \ .$$

The set $[\![\boldsymbol{P}]\!]_{\mathsf{JU}}$ of *JU-models of a DLP* $\boldsymbol{P}$ consists of all stable models J of the program

$$\mathsf{all}(\boldsymbol{P}^{\mathsf{e}}) \setminus \mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}^{\mathsf{e}}, J) \ .$$

Under the JU-semantics, a rule π is rejected if and only if a more recent rule σ is in conflict with π and the body of σ is satisfied in the stable model candidate J. Note that the latter condition is essential – without it, rules might get rejected simply because a more recent rule σ has a conflicting head, without a guarantee that σ will actually be activated.

A related semantics which prevents rejected rules from rejecting other rules is the *update answer sets semantics*, or *AS-semantics* for short (Eiter et al., 2002):

Definition 2.53 (AS-Semantics (Eiter et al., 2002))**.** Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP and J an ASP interpretation. We define the set of rejected rules $\mathsf{rej}_{\mathsf{AS}}(\boldsymbol{P}, J)$ as

$$\mathsf{rej}_{\mathsf{AS}}(\boldsymbol{P}, J) = \left\{ \, \pi \in \mathcal{P}_i \,\middle|\, \exists j \, \exists \sigma : i < j < n \wedge \sigma \in \mathcal{P}_j \setminus \mathsf{rej}_{\mathsf{AS}}(\boldsymbol{P}, J) \wedge \pi \bowtie \sigma \wedge J \models B(\sigma) \, \right\} \ .$$ [8]

The set $[\![\boldsymbol{P}]\!]_{\mathsf{AS}}$ of *AS-models of a DLP* $\boldsymbol{P}$ consists of all stable models J of the program

$$\mathsf{all}(\boldsymbol{P}^{\mathsf{e}}) \setminus \mathsf{rej}_{\mathsf{AS}}(\boldsymbol{P}^{\mathsf{e}}, J) \ .$$

The definitions of the JU- and AS-semantics are fairly straightforward and reflect intuitions about rule updates better than an approach based on a belief update construction, such as (Alferes and Pereira, 1996). As an illustration, let us look at the result of these semantics when applied to Example 1.6:

Example 2.54 (Leite and Pereira (1997))**.** Consider again the program $\mathcal{P}$ from Example 1.6 which contains the rules

$$\mathsf{GoHome} \leftarrow \sim\mathsf{Money}. \qquad \mathsf{GoRestaurant} \leftarrow \mathsf{Money}. \qquad \mathsf{Money}.$$

[8] Note that although this definition is recursive, the defined set is unique. This is because we assume that every rule is uniquely identified and to determine whether a rule from $\mathcal{P}_i$ is rejected, the recursion only refers to rejected rules from programs $\mathcal{P}_j$ with j strictly greater than i. One can thus first find the rejected rules in $\mathcal{P}_n$ (always $\emptyset$ by the definition), then those in $\mathcal{P}_{n-1}$ and so on until $\mathcal{P}_1$.

and its update $\mathcal{U}$ with the rules

$$\sim\mathsf{Money} \leftarrow \mathsf{Robbed}. \qquad\qquad \mathsf{Robbed}.$$

Following the discussion in Example 1.6, the expected stable model of the DLP $\langle \mathcal{P}, \mathcal{U} \rangle$ is $J = \{ \mathsf{Robbed}, \mathsf{GoHome} \}$. Also, $\mathsf{rej}_{\mathsf{JU}}(\langle \mathcal{P}, \mathcal{U} \rangle^{\mathsf{e}}, J) = \mathsf{rej}_{\mathsf{AS}}(\langle \mathcal{P}, \mathcal{U} \rangle^{\mathsf{e}}, J) = \{ \mathsf{Money}. \}$, and J is indeed a stable model of the remaining rules in $\mathsf{all}(\langle \mathcal{P}, \mathcal{U} \rangle^{\mathsf{e}})$. Furthermore, J is the only ASP interpretation with these properties, so

$$[\![\langle \mathcal{P}, \mathcal{U} \rangle]\!]_{\mathsf{JU}} = [\![\langle \mathcal{P}, \mathcal{U} \rangle]\!]_{\mathsf{AS}} = \{ J \} \enspace .$$

Nevertheless, problematic examples which are not handled correctly by these semantics have also been identified (Leite, 2003). Many of them involve tautological updates, the intuition being that a tautological rule (i.e. a rule whose head literal also belongs to its body) cannot indicate a change in the modelled world because it is always true. It thus follows that a tautological update should not affect the stable models of the original program. This is also reflected in the belief update principle (B2.⊤).

An example of a misbehaviour of the AS-semantics is the DLP

$$\boldsymbol{P}_1 = \langle \{ p. \}, \{ \neg p. \}, \{ p \leftarrow p. \} \rangle \qquad \text{with} \qquad [\![\boldsymbol{P}_1]\!]_{\mathsf{AS}} = \{ \{ \neg p \}, \{ p \} \} \enspace . \tag{2.2}$$

where the expected result is a single stable model $\{ \neg p \}$. In this case, the JU-semantics provides a solution: since it allows rejected rules to reject, the initial rule is always rejected and $[\![\boldsymbol{P}_1]\!]_{\mathsf{JU}} = \{ \{ \neg p \} \}$ as expected. Unfortunately, there are also numerous DLPs to which the JU-semantics assigns unwanted models, e.g.

$$\boldsymbol{P}_2 = \langle \{ p. \}, \{ \sim p \leftarrow \sim p. \} \rangle \qquad \text{with} \qquad [\![\boldsymbol{P}_2]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}_2]\!]_{\mathsf{AS}} = \{ \emptyset, \{ p \} \} \enspace . \tag{2.3}$$

The unwanted stable model $\emptyset$ arises because the default assumption $\sim p$ is "reinstated" despite p being initially asserted as a fact. This problem is addressed by constraining the set of default assumptions in the *dynamic stable models semantics*, or *DS-semantics* for short (Alferes et al., 2000):

Definition 2.55 (DS-Semantics (Alferes et al., 2000))**.** Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP and J an ASP interpretation. The set of rejected rules $\mathsf{rej}_{\mathsf{DS}}(\boldsymbol{P}, J)$ is identical to the set $\mathsf{rej}_{\mathsf{JU}}(\boldsymbol{P}, J)$ and we define the set of default assumptions $\mathsf{def}(\boldsymbol{P}, J)$ as

$$\mathsf{def}(\boldsymbol{P}, J) = \{ \sim l \mid l \in \mathcal{L}_{\mathsf{G}} \land \neg \exists \pi \in \mathsf{all}(\boldsymbol{P}) : H(\pi) = \{ l \} \land J \models B(\pi) \} \enspace .$$

The set $[\![\boldsymbol{P}]\!]_{\mathsf{DS}}$ of *DS-models of a DLP* $\boldsymbol{P}$ consists of all ASP interpretations J such that

$$J' = \mathsf{least}([\mathsf{all}(\boldsymbol{P}^{\mathsf{e}}) \setminus \mathsf{rej}_{\mathsf{DS}}(\boldsymbol{P}^{\mathsf{e}}, J)] \cup \mathsf{def}(\boldsymbol{P}^{\mathsf{e}}, J)) \enspace ,$$

where $J' = J \cup \sim(\mathcal{L}_{\mathsf{G}} \setminus J)$ and $\mathsf{least}(\cdot)$ denotes the least model of the argument program with all literals treated as atoms.

Note that it follows from the definition of a (regular) stable model that J is a JU-stable model of a DLP $\boldsymbol{P}$ if and only if

$$J' = \mathsf{least}([\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}_{\mathsf{DS}}(\boldsymbol{P}, J)] \cup \sim(\mathcal{L}_{\mathsf{G}} \setminus J)) \enspace ,$$

Hence the difference between the JU- and DS-semantics is only in the set of default assumptions that can be adopted to construct the model. In particular, if a rule that derives an objective literal l is present in $\mathsf{all}(\boldsymbol{P})$, then $\sim l$ is not among the default assumptions in the DS-semantics although it could be used as a default assumption in the JU semantics. The DS-semantics thus resolves problems with examples such as (2.3), i.e. it holds that $[\![\boldsymbol{P}_2]\!]_{\mathsf{DS}} = \{ \{ p \} \}$. But even the DS-semantics exhibits problematic behaviour when tautological updates are involved, e.g.

$$\boldsymbol{P}_3 = \langle \{ p., \neg p. \}, \{ p \leftarrow p. \} \rangle \quad \text{with} \quad [\![\boldsymbol{P}_3]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}_3]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}_3]\!]_{\mathsf{DS}} = \{ \{ p \} \} \enspace . \tag{2.4}$$

The expected result here is that no stable model should be assigned to $\boldsymbol{P}_3$ because initially it has none and the tautological update should not change anything about that situation. Trouble with tautological and some other types of irrelevant updates has been finally resolved by Alferes et al. (2005) who defined the *refined extension principle* as well as a rule update semantics satisfying the principle. The definition of this semantics is, almost magically, very similar to the DS-semantics, the only difference being that in the set of rejected rules, $i \leq j$ is required instead of $i < j$. The semantics is thus called the *refined dynamic stable models semantics*, or *RD-semantics* for short:

Definition 2.56 (RD-Semantics (Alferes et al., 2005))**.** Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP and J an ASP interpretation. We define the set of rejected rules $\mathsf{rej}_{\mathsf{RD}}(\boldsymbol{P}, J)$ as

$$\mathsf{rej}_{\mathsf{RD}}(\boldsymbol{P}, J) = \{ \pi \in \mathcal{P}_i \mid \exists j \, \exists \sigma : i \leq j < n \land \sigma \in \mathcal{P}_j \land \pi \bowtie \sigma \land J \models B(\sigma) \} \enspace .$$

The set $[\![\boldsymbol{P}]\!]_{\mathsf{RD}}$ of *RD-models of a DLP* $\boldsymbol{P}$ consists of all ASP interpretations J such that

$$J' = \mathsf{least}([\mathsf{all}(\boldsymbol{P}^{\mathsf{e}}) \setminus \mathsf{rej}_{\mathsf{RD}}(\boldsymbol{P}^{\mathsf{e}}, J)] \cup \mathsf{def}(\boldsymbol{P}^{\mathsf{e}}, J)) \enspace ,$$

where J' and $\mathsf{least}(\cdot)$ are as before.

Due to satisfying the refined extension principle, the RD-semantics is completely immune to tautological updates. For instance, in case of example (2.4) we obtain $[\![\boldsymbol{P}_3]\!]_{\mathsf{RD}} = \emptyset$. As we shall see, a vast majority of rule update semantics, even those that have been developed much later and are not based on causal rejection, are not immune to tautological updates.

The rule update semantics introduced above are strongly related to one another. The above considerations show that undesired stable models of the AS-, JU- and DS-semantics were eliminated by enlarging the set of rejected rules or by shrinking the set of default assumptions. The following theorem shows that no additional stable models were added in the process:

Theorem 2.57 (Leite (2003); Alferes et al. (2005))**.** *Let* $\boldsymbol{P}$ *be a DLP. Then,*

$$[\![\boldsymbol{P}]\!]_{\mathsf{AS}} \supseteq [\![\boldsymbol{P}]\!]_{\mathsf{JU}} \supseteq [\![\boldsymbol{P}]\!]_{\mathsf{DS}} \supseteq [\![\boldsymbol{P}]\!]_{\mathsf{RD}} \enspace .$$

Moreover, for each inclusion above there exists a DLP for which the inclusion is strict.

Furthermore, all of these semantics coincide when only acyclic DLPs are considered, showing that the differences in the definitions of rejected rules and default assumptions are only relevant in the presence of cyclic dependencies between literals.

Theorem 2.58 (Homola (2004))**.** *Let* $\boldsymbol{P}$ *be an acyclic DLP. Then,*

$$[\![\boldsymbol{P}]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}]\!]_{\mathsf{DS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RD}} \enspace .$$

The relation between these semantics and other formalisms has also been studied. It has been shown in (Eiter et al., 2002) that the AS-semantics coincides with the non-disjunctive case of the semantics for inheritance programs by Buccafurri et al. (1999). Zhang and Foo (2005) showed how the AS- and DS-semantics can be characterised in terms of *logic program forgetting*.

One of the open issues with these semantics is that one cannot easily *condense* a DLP to a single logic program that could be used *instead* of the DLP to perform further updates. The first obstacle is that the stable models of a DLP may be non-minimal, e.g.

$$\boldsymbol{P}_4 = \langle \{ p., q \leftarrow p. \}, \{ \sim p \leftarrow \sim q. \} \rangle \qquad \text{with} \qquad [\![\boldsymbol{P}_4]\!]_{\mathsf{RD}} = \{ \emptyset, \{ p, q \} \} \enspace .$$

Since stable models of non-disjunctive programs are subset-minimal, no such program can have the set of stable models $[\![\boldsymbol{P}_4]\!]_{\mathsf{RD}}$. Condensing to a disjunctive program is also problematic because rule update semantics are constrained to non-disjunctive programs only, so after a condensation one would not be able to perform any further updates.

Nevertheless, for each of the four discussed semantics (AS-, JU-, DS- and RD-semantics), there do exist translations of a DLP to a single non-disjunctive program *over an extended language* whose stable models correspond one-to-one with the stable models assigned to the DLP under the respective rule update semantics. Due to the language extension, the new program cannot simply be updated directly as a substitute for the original DLP, but may serve as a way to study the computational properties of the rule update semantics and as a way to implement it using existing answer-set solvers.

Related to this are also the semantics proposed by Osorio and Cuevas (2007) as simpler substitutes for the AS-semantics. Unlike the semantics discussed above, they are not defined declaratively, instead they are specified directly by translating the initial program and its update to a single program *over the same language*. The first translation essentially weakens the rules from the original program by making them defeasible, i.e. a rule $l \leftarrow B(\pi)$ is transformed into the rule $l \leftarrow B(\pi), \sim\bar{l}$. The authors have shown that the resulting semantics is equivalent to the AS-semantics if a single update is performed and the updating program contains a tautology $l \leftarrow l$ for every objective literal l. Due to its simplicity, this semantics is *not* sensitive to the addition and removal of tautologies, but it adopts the problematic behaviour of the AS-semantics even when the tautologies are *removed* from the updating program. For example, when considering the DLPs

$$\boldsymbol{P}_5 = \langle \{ p., \neg p. \}, \emptyset \rangle \qquad \text{and} \qquad \boldsymbol{P}'_5 = \langle \{ p., \neg p. \}, \{ p \leftarrow p., \neg p \leftarrow \neg p. \} \rangle$$

the AS-semantics correctly assigns no model to $\boldsymbol{P}_5$ although its sensitivity to tautological updates causes $\boldsymbol{P}'_5$ to have two AS-models: $\{ p \}$ and $\{ \neg p \}$. The first semantics suggested by Osorio and Cuevas (2007) assigns these two stable models to both $\boldsymbol{P}_5$ and $\boldsymbol{P}'_5$, so it exhibits problematic behaviour even on DLPs that were correctly handled by the AS-semantics.

The second translation is more involved as it produces a program that may not be expressible by a non-disjunctive program. The resulting update semantics is shown to coincide with the AS-semantics in case only a single update is performed. Note that since the AS-semantics coincides with the JU-semantics on DLPs of length two, the above mentioned relationships between semantics from (Osorio and Cuevas, 2007) and the AS-semantics also hold for the JU-semantics. Their behaviour on DLPs of length three or more has not been studied.

2.8.2 Preference-Based Semantics

A smaller group of rule update semantics relies on syntactic transformations and semantics for *prioritised logic programs*. In these semantics, default negation in heads of rules is not considered.

Formally, a prioritised logic program is a pair $(\mathcal{P}, \prec)$ where $\mathcal{P}$ is a program and $\prec$ is a strict partial order over $\mathcal{P}$. The intuitive meaning of $\prec$ is that if $\pi \prec \sigma$, then σ is more preferred than π. There exist a number of different semantics for prioritised logic programs. Their goal is to plausibly use the preference relation $\prec$ to choose the preferred stable models among the stable models of $\mathcal{P}$.

One rule update semantics of this type was defined by Zhang (2006) and relies on the semantics for prioritised logic programs from (Zhang, 2003).[9] Generally speaking, the semantics is assigned to a prioritised logic program $(\mathcal{P}, \prec)$ by pruning away less preferred rules, obtaining an ordinary logic program $\mathcal{P}^{\prec} \subseteq \mathcal{P}$ called a *reduct*. A prioritised logic program may have zero or more reducts and the preferred stable models are all the stable models of all the reducts. For a detailed discussion of reducts and their properties the reader can refer to (Zhang, 2003).

Subsequently, the update semantics defined in (Zhang, 2006) performs an update of a program $\mathcal{P}$ by a program $\mathcal{U}$ by performing the following steps:

1. Take some stable model $J_{\mathcal{P}}$ of $\mathcal{P}$.
2. Update $J_{\mathcal{P}}$ by $\mathcal{U}$ using a specialised semantics for performing interpretation updates, somewhat similar to (Marek and Truszczynski, 1998) but specified using a prioritised logic program. Denote some ASP interpretation resulting from this update by $J_{\langle \mathcal{P}, \mathcal{U} \rangle}$.
3. Extract a maximal subset $\mathcal{P}'$ of $\mathcal{P}$ that is coherent with $J_{\langle \mathcal{P}, \mathcal{U} \rangle}$.[10]
4. The set of reducts of the prioritised logic program $(\mathcal{P}' \cup \mathcal{U}, \mathcal{P}' \times \mathcal{U})$ is the result of updating $\mathcal{P}$ by $\mathcal{U}$.

As explained in (Zhang, 2006), the intuition behind the first two steps is that simply taking a maximal subset of $\mathcal{P}$ coherent with $\mathcal{U}$ is too crude an operation because it does not take into account the source of a conflict.

Example 2.59 (Intuition For Steps 1. and 2. (Zhang, 2006))**.** Consider the programs

$$\mathcal{P}: \quad p. \qquad\qquad \text{and} \qquad\qquad \mathcal{U}: \quad r \leftarrow p.$$
$$\qquad q \leftarrow r. \qquad\qquad\qquad\qquad\qquad \neg q \leftarrow r.$$

Since $\mathcal{P} \cup \mathcal{U}$ is incoherent, some part of $\mathcal{P}$ needs to be eliminated to regain coherence. There are two maximal subsets of $\mathcal{P}$ that are coherent with $\mathcal{U}$: $\{ p. \}$ and $\{ q \leftarrow r. \}$. However, intuition suggests that the former set is preferable since the direct conflict between rules $(q \leftarrow r.)$ and $(\neg q \leftarrow r.)$ provides a justification for eliminating the rule $(q \leftarrow r.)$ and thus keeping the fact $(p.)$.

The approach taken, then, is to first take a stable model of $\mathcal{P}$ and update it by $\mathcal{U}$, obtaining a new ASP interpretation $J_{\langle \mathcal{P}, \mathcal{U} \rangle}$ that reflects the new information in $\mathcal{U}$. Afterwards, a maximal set of rules from $\mathcal{P}$ coherent with $J_{\langle \mathcal{P}, \mathcal{U} \rangle}$ is used to form a prioritised logic program that prefers rules from $\mathcal{U}$ over rules from $\mathcal{P}$. The reducts of this program form the result of the update.

[9]Note that the preference relation in these papers is reversed w.r.t. the one we use in this thesis, i.e. $\pi \prec \sigma$ means in (Zhang, 2003, 2006) that π is more preferred than σ.

[10]That is, a maximal $\mathcal{P}' \subseteq \mathcal{P}$ such that there exists a stable model of $\mathcal{P}' \cup \{ l. \mid l \in J_{\langle \mathcal{P}, \mathcal{U} \rangle} \}$.

Due to the possibility of having multiple reducts as possible results of the update, it is not completely clear how updates can be iterated. Do we choose one reduct and commit to it? Which one do we choose, then? Or do we simply consider all of the reducts and all possible evolutions? Due to these unresolved issues, we formally define this semantics only for DLPs of length two. We call it *preference-based Zhang's semantics*, or *PRZ-semantics* for short.

Definition 2.60 (PRZ-Semantics (Zhang, 2006))**.** Let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$ be a DLP without default negation in heads of rules. The set $[\![\boldsymbol{P}]\!]_{\mathsf{PRZ}}$ of *PRZ-models of* $\boldsymbol{P}$ is the union of sets of stable models of all reducts obtained by performing the steps 1. – 4. above.

One distinguishing feature of the PRZ-semantics is that by relying on a stable model of $\mathcal{P}$ for conflict resolution, it is unable to detect "latent" conflicts between rules that have not been "triggered" in the initial stable model or its update. This is illustrated in the following example:[11]

Example 2.61 (Undetected Latent Conflicts in (Zhang, 2006))**.** Consider the programs

$$\mathcal{P}: \begin{array}{l} p \leftarrow r. \\ q \leftarrow r. \end{array} \qquad \text{and} \qquad \mathcal{U}: \begin{array}{r} r. \\ \neg p \leftarrow q. \end{array}$$

and let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$. The single stable model of $\mathcal{P}$ is $J_{\mathcal{P}} = \emptyset$ and its update by $\mathcal{U}$ results in the interpretation $J_{\langle \mathcal{P}, \mathcal{U} \rangle} = \{ r \}$ which is coherent with $\mathcal{P}$. The resulting prioritised logic program $(\mathcal{P} \cup \mathcal{U}, \mathcal{P} \times \mathcal{U})$ has only one reduct, $\mathcal{P} \cup \mathcal{U}$, that has no stable model. In other words, $[\![\boldsymbol{P}]\!]_{\mathsf{PRZ}} = \emptyset$ and the conflict between $\mathcal{P}$ and $\mathcal{U}$ remained unresolved. Note also that $[\![\boldsymbol{P}]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}]\!]_{\mathsf{DS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RD}} = \{ \{ \neg p, q, r \} \}$.

The semantics from (Zhang, 2006) is also sensitive to tautological updates:

Example 2.62 (Tautological Updates in (Zhang, 2006))**.** Consider the programs

$$\mathcal{P}: \begin{array}{r} p \leftarrow {\sim}\neg p. \\ \neg p \leftarrow {\sim} p. \end{array} \qquad \text{and} \qquad \mathcal{U}: \; p \leftarrow p.$$

Both stable models $\{ p \}$ and $\{ \neg p \}$ of $\mathcal{P}$ remain unchanged after an update by $\mathcal{U}$ and thus both rules of $\mathcal{P}$ are retained in the resulting prioritised logic program $(\mathcal{P} \cup \mathcal{U}, \mathcal{P} \times \mathcal{U})$. Its only reduct, however, is the program $\{ p \leftarrow {\sim}\neg p., p \leftarrow p. \}$, which has a single stable model $\{ p \}$. The tautological update has thus discarded one of the stable models of $\mathcal{P}$.

It also follows from the results of (Zhang, 2006) that the computational complexity of the PRZ-semantics is higher than that of the causal rejection-based semantics.

Preference-based rule update semantics were also considered by Delgrande et al. (2007), utilising the semantics for prioritised logic programs examined in (Schaub and Wang, 2003).[12] Instead of defining how a prioritised logic program $(\mathcal{P}, \prec)$ can be characterised in terms of reducts, as in (Zhang, 2003), Schaub and Wang (2003) specify conditions that a stable model of $\mathcal{P}$ must satisfy in order to be a *preferred stable model of* $(\mathcal{P}, \prec)$. They use three such conditions, defined in the literature on programs with preferences,

[11] This example does not apply to an earlier version of the PRZ-semantics from (Zhang and Foo, 1998). This is because the maximal subset $\mathcal{P}'$ of $\mathcal{P}$ chosen for constructing the prioritised logic program is required to be coherent with both $J_{\langle \mathcal{P}, \mathcal{U} \rangle}$ and $\mathcal{U}$, not only with $J_{\langle \mathcal{P}, \mathcal{U} \rangle}$ as in (Zhang, 2006).

[12] Prioritised logic programs are called *ordered logic programs* in (Schaub and Wang, 2003; Delgrande et al., 2007).

dubbed *D-preference*, *W-preference* and *B-preference*, which yield an increasing number of preferred stable models. For further details about these preference strategies the reader can refer to (Schaub and Wang, 2003) and the references therein.

Unlike in (Zhang, 2006), the methodology chosen for performing rule updates is based on relatively simple transformations into a prioritised logic program. In order to define these transformations, we first need to introduce the following notation for arbitrary programs $\mathcal{P}$ and $\mathcal{U}$:

$$\begin{aligned}
\mathcal{P}^d &= \{\, l \leftarrow B(\pi), {\sim}\bar{l}. \mid (l \leftarrow B(\pi).) \in \mathcal{P} \,\} \\
C(\mathcal{P},\mathcal{U}) &= \{\, (\pi,\sigma) \mid \exists l \in \mathcal{L}_{\mathsf{G}} : \pi \in \mathcal{P} \wedge \sigma \in \mathcal{U} \wedge H(\pi) = \{\, l \,\} \wedge H(\sigma) = \{\, \bar{l} \,\} \,\} \\
c(\mathcal{P},\mathcal{U}) &= \{\, \pi, \sigma \mid (\pi,\sigma) \in C(\mathcal{P},\mathcal{U}) \,\}
\end{aligned}$$

Intuitively, $\mathcal{P}^d$ denotes a program obtained from $\mathcal{P}$ by making all its rules defeasible, analogically to the semantics based on weakenings by Osorio and Cuevas (2007). The set $C(\mathcal{P},\mathcal{U})$ contains pairs of rules from $\mathcal{P}$ and $\mathcal{U}$ with conflicting heads and $c(\mathcal{P},\mathcal{U})$ contains rules from $\mathcal{P}$ and $\mathcal{U}$ involved in such conflicts.

Delgrande et al. (2007) proposed three different operators for updating a program $\mathcal{P}$ by a program $\mathcal{U}$, each of which outputs a different prioritised logic program:

$$\begin{aligned}
\mathcal{P} *_0 \mathcal{U} &= \left(\mathcal{P}^d \cup \mathcal{U}^d, \mathcal{P}^d \times \mathcal{U}^d\right) \\
\mathcal{P} *_1 \mathcal{U} &= \left(\mathcal{P}^d \cup \mathcal{U}^d, C(\mathcal{P}^d,\mathcal{U}^d)\right) \\
\mathcal{P} *_2 \mathcal{U} &= \left(c(\mathcal{P},\mathcal{U})^d \cup ((\mathcal{P} \cup \mathcal{U}) \setminus c(\mathcal{P},\mathcal{U})), C(\mathcal{P}^d,\mathcal{U}^d)\right)
\end{aligned}$$

Informally, $*_0$ makes all rules from $\mathcal{P}$ and $\mathcal{U}$ defeasible and gives preference to every rule from $\mathcal{U}$ over any rule from $\mathcal{P}$. The operator $*_1$ produces a more cautious preference relation, only preferring rules from $\mathcal{U}$ over rules from $\mathcal{P}$ with conflicting heads. In addition, the operator $*_2$ refrains from weakening rules that are not involved in any conflict.

It is argued in (Delgrande et al., 2007) that these operators can be naturally generalised to account for arbitrary (finite) sequences of programs as follows:

$$*(\langle \mathcal{P}_i \rangle_{i<n}) = \begin{cases} \mathcal{P}_0 * \mathcal{P}_1 & \text{if } n = 2 \\ *(\langle \mathcal{P}_i \rangle_{i<n-1}) * \mathcal{P}_{n-1} & \text{if } n > 2 \end{cases}$$

Nevertheless, this definition is slightly incomplete since the result of operators $*_0$, $*_1$, $*_2$ is not an ordinary logic program but a prioritised one. The question then arises as to what happens with the priority relation of an intermediate result, say $\mathcal{P}_0 * \mathcal{P}_1$, when it is further updated by $\mathcal{P}_2$. Is it going to be discarded or merged with the new preference relation? In either case, the result should be specified by the definition.

In the following we assume that the preference relations are merged and measures are taken to ensure that the merged relation remains a strict partial order, i.e. transitivity is enforced after the merge. We can now define the update semantics of (Delgrande et al., 2007) for arbitrary DLPs. We call them the *PRX$_i$-semantics* with X representing the preference strategy (i.e. X is one of D, W, B) and i denoting the particular operator used for forming the prioritised logic program (i.e. $i \in \{\, 0, 1, 2 \,\}$).

Definition 2.63 (PRX$_i$-Semantics (Delgrande et al., 2007))**.** Let $\boldsymbol{P}$ be a DLP without default negation in heads of rules, X be one of D, W, B and $i \in \{\, 0, 1, 2 \,\}$. The set $[\![\boldsymbol{P}]\!]_{\mathsf{PRX}_i}$ of *PRX$_i$-models of* $\boldsymbol{P}$ is the set of preferred stable models of the prioritised logic program

$*_i(\boldsymbol{P})$ under the preference strategy X.

The overall properties of these rule update semantics depend on the chosen operator ($*_0$, $*_1$ or $*_2$) and on the chosen preference strategy (D-, W- or B-preference). Nevertheless, as shown on examples in (Delgrande et al., 2007), all PRX_i-semantics are sensitive to tautological updates. The following example shows an interesting behaviour that distinguishes these semantics from the previously discussed ones:

Example 2.64 (Default Assumptions vs. Facts in (Delgrande et al., 2007))**.** Consider the programs

$$\mathcal{P}: \quad \neg p. \qquad\qquad \text{and} \qquad\qquad \mathcal{U}: \quad p \leftarrow \sim\neg p.$$

and let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$. For any operator $*_i$ and preference strategy X, $[\![\boldsymbol{P}]\!]_{\mathsf{PRX_i}} = \{\{p\}\}$. This indicates that the default assumption in the updating program is given preference over the fact in the initial program. If we interpret p as $\mathsf{Man}(mary)$, then this example shows that if initially $\mathsf{Man}(mary)$ is known to be false and later we learn that

$$\mathsf{Man}(\mathbf{x}) \leftarrow \sim\neg\mathsf{Man}(\mathbf{x}). \ ,$$

meaning that by default all individuals are men, then this immediately changes our knowledge about $\mathsf{Man}(mary)$: we now know that $\mathsf{Man}(mary)$ is true!

It seems more natural to give preference to initial facts over default assumptions in more recent rules. Note that $[\![\boldsymbol{P}]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}]\!]_{\mathsf{JU}} = \{\{p\}, \{\neg p\}\}$ and $[\![\boldsymbol{P}]\!]_{\mathsf{DS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RD}} = [\![\boldsymbol{P}]\!]_{\mathsf{PRZ}} = \{\{\neg p\}\}$, i.e. the causal rejection semantics with unrestricted set of default assumptions allow both $\{p\}$ and $\{\neg p\}$ to be stable models of $\boldsymbol{P}$ while the "fixed" versions of these semantics together with Zhang's preference-based semantics actually prefer the initial fact over the default assumption.

2.8.3 Other Approaches

Sakama and Inoue (2003) have proposed a rule update semantics that is clearly based on ideas from belief revision, similarly as formula-based belief update operators. In particular, they define that a program $\mathcal{P}' \cup \mathcal{U}$ achieves the update of $\mathcal{P}$ by $\mathcal{U}$ if $\mathcal{P}'$ is a maximal subset of $\mathcal{P}$ such that $\mathcal{P}' \cup \mathcal{U}$ has a stable model.

As with the PRZ-semantics, we define the semantics of Sakama and Inoue (2003) only for DLPs of length two because it is not clear how one should deal with multiple results of an update. We call the resulting semantics the *RVS-semantics*:

Definition 2.65 (RVS-Semantics (Sakama and Inoue, 2003))**.** Let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$ be a DLP. The set $[\![\boldsymbol{P}]\!]_{\mathsf{RVS}}$ of *RVS-models of* $\boldsymbol{P}$ is the union of sets of stable models of all programs $\mathcal{P}' \cup \mathcal{U}$ where $\mathcal{P}'$ is a maximal subset of $\mathcal{P}$ such that $\mathcal{P}' \cup \mathcal{U}$ is coherent.

As discussed in Example 2.59, the approach adopted by the RVS-semantics pays no attention to the source of conflicts – any solution of a conflict is as good as any other as long as only a minimal set of rules is eliminated. Another consequence is that conflicts are removed at any cost, even if there is no plausible way to explain why the update should restore coherence. This has been criticised in (Leite, 2003), arguing that every conflict has several causes and each type of conflict should be dealt with accordingly. One consequence of this is that an empty or tautological update may restore coherence (and consistency) of an initial program. If we compare this to belief change principles and operators, such a behaviour is typical of *revision* but is not desirable for *updates*.

Similar ideas form the basis of the rule *revision* semantics proposed by Delgrande (2010). Note that since the distinction between program update and revision, as these terms are used in the literature, is somewhat blurry, in the following we also refer to this semantics as an *update semantics*. Informally, the stable model of a sequence of programs is constructed by first keeping all rules from the last program and committing to a minimal set of default literals used to derive one of its stable models. Subsequently, a maximal coherent subset of the previous program is added and further commitments are made. This process is iterated until the first program of the sequence is processed. For illustration, consider the DLP

$$\boldsymbol{P} = \langle \{ p. \} , \{ q. \} , \{ r \leftarrow \sim p., r \leftarrow \sim q. \} \rangle \enspace .$$

We start with the last program of the sequence which has the stable model $\{ r \}$. This stable model can be derived either using the literal $\sim p$ or $\sim q$ and we need choose one of these and commit to it. If we pick the former, the overall set of literals we commit to at this stage is $\{ \sim p, r \}$. We then proceed to the second program and realise that it is coherent with our commitments as well as the rules from the last program. We thus add q to our set of commitments and the fact $(q.)$ to the set of rules that we are going to keep. Proceeding to the first program of the sequence, the rule within it is inconsistent with our commitment to $\sim p$, so it needs to be discarded. The set of objective literals we committed to until now, namely $\{ q, r \}$, forms one stable model of $\boldsymbol{P}$. Note that if we initially commit to $\sim q$, we obtain the stable model $\{ p, r \}$.

A reader interested in the technical subtleties of this semantics is encouraged to consult (Delgrande, 2010). In this thesis we refer to this semantics as the *RVD-semantics* and constrain ourselves to the following, rather informal definition.

Definition 2.66 (RVD-Semantics (Delgrande, 2010))**.** Let $\boldsymbol{P}$ be a DLP. The set $[\![\boldsymbol{P}]\!]_{\mathsf{RVD}}$ of *RVD-models of* $\boldsymbol{P}$ is the set of answer sets of $\boldsymbol{P}$ as defined in Definition 4.4 of (Delgrande, 2010).

Similarly as the RVS-semantics, the RVD-semantics resolves conflicts at any cost, a consequence of which is that empty and tautological updates restore coherence and consistency. Furthermore, it exhibits the same behaviour as the PRX_i-semantics in Example 2.64, i.e. it prefers to satisfy default assumptions in further programs to satisfying earlier facts. It actually goes even further than the PRX_i-semantics, as illustrated in the following example:

Example 2.67 (Default Assumptions vs. Facts in (Delgrande, 2010))**.** Consider the programs

$$\mathcal{P} : \quad p. \qquad\qquad \text{and} \qquad\qquad \mathcal{U} : \quad q \leftarrow \sim p.$$

and let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$. We obtain $[\![\boldsymbol{P}]\!]_{\mathsf{RVD}} = \{ \{ q \} \}$, as opposed to $[\![\boldsymbol{P}]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}]\!]_{\mathsf{DS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RD}} = [\![\boldsymbol{P}]\!]_{\mathsf{PRZ}} = [\![\boldsymbol{P}]\!]_{\mathsf{PRX_i}} = [\![\boldsymbol{P}]\!]_{\mathsf{RVS}} = \{ \{ p \} \}$. This indicates that the default assumptions in the updating program are given preference over facts from the initial program even more aggressively than in case of the PRX_i-semantics. If we interpret p as $\mathsf{Dog}(gordo)$ and q as $\neg\mathsf{CanBark}(gordo)$, then this example shows that if initially $\mathsf{Dog}(gordo)$ is known to be true and later we learn that

$$\neg\mathsf{CanBark}(\mathbf{x}) \leftarrow \sim\mathsf{Dog}(\mathbf{x}). \enspace ,$$

meaning that, by default, individuals that are not dogs can't bark, then this immediately

Table 2.8: Applicability of rule update semantics

Semantics	Applicability
AS, JU, DS, RD	Arbitrary DLPs.
PRX_i, RVD	DLPs without default negation in heads of rules.
PRZ, RVS	DLPs of length two without default negation in heads of rules.

modifies our knowledge about $gordo$: we no longer know whether $\mathsf{Dog}(gordo)$ is true or not and, in addition, we conclude that $\mathsf{CanBark}(gordo)$ is (explicitly) false.

A methodology based on maximal subsets of the initial program coherent with its update was also used by Osorio and Zepeda (2007) for updating programs under the *pstable models semantics*. The idea is used indirectly by augmenting the bodies of original rules with additional literals and using an abductive framework to minimise the set of rules "disabled" by falsifying the added literal. We do not further consider this semantics because it diverges from the standard notion of a stable model and uses pstable models instead.

Finally, there also exist approaches based on a semantic framework that directly encodes literal dependencies induced by rules, and performs changes on the dependencies instead of on the rules themselves. The advantage over dealing with rules is that the dependency relation is monotonic, so AGM postulates and operators can be applied to it directly (Krümpelmann and Kern-Isberner, 2010; Krümpelmann, 2012). In the work of Šefránek (2006, 2011), the dependency framework is used for specifying *irrelevant updates*, an instance of which are tautological updates, and designing update semantics immune to such irrelevant updates.

2.8.4 Fundamental Properties

As demonstrated above, rule update semantics are based on a number of different approaches and constructions and provide different results even on very simple examples. But the goal of this thesis is to address updates of hybrid knowledge bases which subsume logic programs. Any resulting hybrid update framework, when applied only to rules, induces a corresponding rule update semantics. So from the point of view of finding a *suitable hybrid update framework*, the fundamental properties that rule update semantics have *in common* are more important than the differences between them.

In this section we indicate and examine five such properties. We call them *syntactic* because their formulation requires that we refer to the syntax of the respective DLP. The first three properties are satisfied by all rule update semantics that we formally introduced above. The other two are satisfied by semantics based on causal rejection and thus serve as entry points for comparing any rule update semantics with all causal rejection-based semantics simultaneously.

Note that the distinct semantics have been defined for different classes of DLPs. When we say that a semantics S satisfies a particular property, we constrain ourselves only to DLPs that it can handle as inputs. The classes of DLPs to which the introduced semantics are applicable is summarised in Table 2.8.

The first fundamental property captures the fact that rule update semantics produce only *supported* models. In a static setting, support (Apt et al., 1988; Dix, 1995b) is one of the basic conditions that Logic Programming semantics are intuitively designed to satisfy. Its generalisation to the dynamic case is straightforward.

Definition 2.68 (Support)**.** Let S be a rule update semantics, $\mathcal{P}$ a program, l an objective literal and J an ASP interpretation. We say that

- $\mathcal{P}$ *supports* l *in* J if for some rule $\pi \in \mathcal{P}$, $l \in H(\pi)$ and $J \models B(\pi)$;
- $\mathcal{P}$ *supports* J if every objective literal $l \in J$ is supported by $\mathcal{P}$ in J;
- *S respects support* if for every DLP $\boldsymbol{P}$ to which S is applicable and every S-model J of $\boldsymbol{P}$, all($\boldsymbol{P}$) supports J.

In other words, a rule update semantics respects support if every objective literal l, true in a stable model assigned to some DLP, is in the head of some rule whose body is true in the same model. Such a rule then provides a *justification* for l.

A consequence of support is that the rule update semantics satisfies a counterpart of *language conservation*, as formulated in Definition 2.42 for first-order update operators. In the context of rule updates it can be defined as follows:

Definition 2.69 (Language Conservation for Rule Updates)**.** Let S be a rule update semantics. We say that *S conserves the language* if for all sets of predicate symbols A, every DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ to which S is applicable and every S-model J of $\boldsymbol{P}$, if $\mathsf{pr}(\mathcal{P}_i) \subseteq A$ for all $i < n$, then $\mathsf{pr}(J) \subseteq A$.

Though support and language conservation are very basic requirements, and certainly too weak to be sufficient for a "good" rule update semantics, they seem to be very intuitive from the Logic Programming perspective. And, indeed, they are satisfied by all rule update semantics that we introduced previously.

Theorem 2.70 (Respect for Support and Language Conservation)**.** *Let X be one of D, W, B and $i \in \{ 0, 1, 2 \}$. The rule update semantics AS, JU, DS, RD, PRZ, PRX$_i$, RVS and RVD respect support and conserve the language.*

Proof (sketch). See Appendix A, page 186. □

The third fundamental property for rule update semantics expresses the usual expectation regarding how facts should be updated by newer facts. An analogical property also holds for Winslett's first-order update operator (c.f. Theorem 2.45), so this case can be seen as the common ground for both ontology and rule updates.

Definition 2.71 (Fact Update)**.** Let S be a rule update semantics. We say that *S respects fact update* if for every finite sequence of consistent sets of facts $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ to which S is applicable, the unique S-model of $\boldsymbol{P}$ is the ASP interpretation

$$\{ l \in \mathcal{L}_{\mathsf{G}} \mid \exists j < n : (l.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow \{ \bar{l}., \sim l. \} \cap \mathcal{P}_i = \emptyset) \} \enspace .$$

Fact update enforces literal inertia, which forms the basis for belief update operators such as Winslett's or Forbus', but only for the case when both the initial program and its updates are consistent sets of facts. Similarly as before, all rule update semantics adhere to this property.

Theorem 2.72 (Respect for Fact Update)**.** *Let X be one of D, W, B and $i \in \{ 0, 1, 2 \}$. The rule update semantics AS, JU, DS, RD, PRZ, PRX$_i$, RVS and RVD respect fact update.*

Proof (sketch). See Appendix A, page 187. □

The fourth and fifth syntactic properties are fundamental for all semantics based on causal rejection. The first of them is the causal rejection principle itself.

Definition 2.73 (Causal Rejection)**.** Let S be a rule update semantics. We say that *S respects causal rejection* if for every DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ to which S is applicable, every S-model J of $\boldsymbol{P}$, all $i < n$ and all rules $\pi \in \mathcal{P}_i$,

$$J \not\models \pi \qquad \text{implies} \qquad \exists j \, \exists \sigma : i < j < n \land \sigma \in \mathcal{P}_j^{\mathsf{e}} \land \pi \bowtie \sigma \land J \models B(\sigma) \ .$$

This principle requires a *cause* for every violated rule in the form of a more recent rule with a conflicting head and a satisfied body. It is hard-wired in the definitions of sets of rejected rules of the four rule update semantics that are based on it.

Theorem 2.74 (Respect for Causal Rejection)**.** *The rule update semantics AS, JU, DS and RD respect causal rejection.*

Proof (sketch). See Appendix A, page 188. □

The final syntactic property stems from the fact that all rule update semantics based on causal rejection coincide on acyclic DLPs (Homola, 2004; Alferes et al., 2005). Thus, the behaviour of any rule update semantics on acyclic DLPs can be used as a way to compare it to all these semantics simultaneously.

Definition 2.75 (Acyclic Justified Update)**.** Let S be a rule update semantics. We say that *S respects acyclic justified update* if for every acyclic DLP $\boldsymbol{P}$ to which S is applicable, the set of S-stable models of $\boldsymbol{P}$ coincides with $[\![\boldsymbol{P}]\!]_{\mathsf{JU}}$.

Theorem 2.76 (Respect for Acyclic Justified Update)**.** *The rule update semantics AS, JU, DS and RD respect acyclic justified update.*

Proof (sketch). See Appendix A, page 188. □

Part II

Updates of MKNF Knowledge Bases

3

Dynamic MKNF Knowledge Bases with Static Rules

JOÃO: So, how's your work going?

MARTIN: Eh... well, it's pretty frustrating. Belief update just doesn't fit with rule updates at all! When I try to combine them, I lose all intuition about what the result should actually do...

JOÃO: Have you tried simplifying the problem a bit?

MARTIN: Hmm... I guess not.

JOÃO: Maybe you should.

SEPTEMBER 2009, LISBON, PORTUGAL

Now that all necessary technical background has been covered, we are ready to start addressing the problem of updating hybrid knowledge bases. From a theoretical viewpoint our objective is to identify a general-purpose update semantics for MKNF knowledge bases. More specifically, we would like to assign some sort of *dynamic MKNF models* to any pair of MKNF knowledge bases representing the original knowledge base and its update. It would also be interesting to do the same for sequences of MKNF knowledge bases, similarly as with belief and rule updates. We thus introduce the following concept:

Definition 3.1 (Dynamic MKNF Knowledge Base)**.** A finite sequence of MKNF knowledge bases is called a *dynamic MKNF knowledge base* (or *DMKB* for short).

In this chapter we address a simplified version of this general problem by constraining ourselves to updates of the ontology component of MKNF knowledge bases while the rule component remains static. In other words, we assume that rules in a hybrid knowledge base remain static or change infrequently so they can be kept up to date by manual editing. Though this restricts the applicability of the resulting update semantics, it still encompasses many practical applications of hybrid knowledge bases, particularly those where the ontology contains highly dynamic information and rules represent default preferences or behaviour that can be overridden by ontology updates when necessary. It allows for reasoning with assumptions and naturally expressing exceptions and provides a seamless two-way interaction between the ontology and rules.

From a more technical viewpoint, the proposed hybrid update semantics is obtained by generalising an immediate consequence operator that characterises MKNF models of MKNF knowledge bases. The operator is augmented with a first-order update operator that takes care of ontology updates and then used to define dynamic MKNF models of DMKBs with static rules. The semantics is thus *parametrised* by a first-order update operator; we particularly consider Winslett's operator as one way to instantiate the semantics and show how it behaves on a simple example.

To the best of our knowledge, this is the first proposal of an update semantics for hybrid knowledge bases. Its overall theoretical properties depend on the properties of the adopted first-order update operator. Assuming that the operator satisfies some of the basic update postulates, we show that the derived hybrid update semantics enjoys a number of desirable properties, namely it

- is faithful to the (static) semantics of MKNF knowledge bases;
- is faithful to the first-order update operator that it is based on;
- adheres to the principle of primacy of new information;
- is immune to tautological updates, i.e. such updates do not affect the semantics.
- is syntax-independent w.r.t. the original ontology and its updates;

Similarly as in most existing work on rule updates, we assume that all MKNF rules are ground and non-disjunctive. In addition, we do not allow for rules with empty heads and for rules with default negation in their heads. These restrictions do not remove any further expressivity: an empty head is interchangeable with the head $\{ \bot \}$ and generalised default literals can be eliminated from non-disjunctive rule heads as described in Proposition 2.22. In the rest of this chapter we implicitly work under these assumptions.

The remainder of this chapter is structured as follows: In Section 3.1 we define the consequence operator for characterising MKNF models of MKNF knowledge bases while in Section 3.2 we imbue this operator with the ability to perform ontology updates and use it to define an update semantics for DMKBs with static rules. We formulate and prove the basic formal properties of this framework in Section 3.3 and conclude in Section 3.4.

The relevant proofs can be found in Appendix B. Preliminary versions of this work have been published in (Slota and Leite, 2010a,b). Differently from those papers, the semantics presented here is not limited to using Winslett's operator for performing ontology updates – any member of a large class of first-order update operators can be used for this purpose.

3.1 Static Consequence Operator

We begin by showing how the semantics of MKNF knowledge bases can be characterised in terms of a consequence operator. The basic idea is very similar to the usual two-valued immediate consequence operator from Logic Programming: given a positive MKNF program, the operator returns the heads of all MKNF rules that are activated by the argument interpretation. This idea can be generalised to deal with positive MKNF knowledge bases by joining the heads of active rules with the translation of the ontology and returning the set of models of the resulting first-order theory. Formally:

Definition 3.2 (Consequence Operators)**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a positive MKNF knowledge base. The *immediate consequence operators* $T_{\mathcal{P}}$ *and* $T_{\mathcal{K}}$ are defined for all $\mathcal{M} \in \mathfrak{M}$ as follows:

$$T_{\mathcal{P}}(\mathcal{M}) = \bigcup \{ H(\pi) \mid \pi \in \mathcal{P} \land \mathcal{M} \models \kappa(B(\pi)) \} \enspace ,$$
$$T_{\mathcal{K}}(\mathcal{M}) = [\![T_{\mathcal{P}}(\mathcal{M}) \cup \kappa(\mathcal{O})]\!] \enspace .$$

Note that an analogous consequence operator operating on sets of modal atoms was presented in (Motik and Rosati, 2010).

We can view the set $\mathfrak{M}$ of all MKNF interpretations together with the empty set as a complete lattice with the greatest element $\mathfrak{I}$.[1] The following example illustrates how the operator can be iterated, starting from the greatest element $\mathfrak{I}$, until a fixed point is reached which coincides with the MKNF model of the MKNF knowledge base.

Example 3.3 (Iterating the Consequence Operators)**.** Consider the MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ where[2]

$$\mathcal{O}: \quad p \lor \neg q \qquad\qquad \mathcal{P}: \quad \begin{aligned} &q \leftarrow r. \\ &q \leftarrow s. \\ &r. \end{aligned}$$

Starting from the interpretation $\mathcal{M}_0 = \mathfrak{I}$, we will iterate the operator $T_{\mathcal{K}}$ until we reach a fixed point. Since $\mathcal{M}_0$ does not satisfy the body of the first two rules in $\mathcal{P}$ but trivially satisfies the body of the last one, it follows that $T_{\mathcal{P}}(\mathcal{M}_0) = \{ r \}$ and the first application of $T_{\mathcal{K}}$ can determined as follows:

$$T_{\mathcal{K}}(\mathcal{M}_0) = [\![T_{\mathcal{P}}(\mathcal{M}_0) \cup \mathcal{O}]\!] = [\![\{ r, p \lor \neg q \}]\!] = \mathcal{M}_1 \enspace .$$

We can now see that $\mathcal{M}_1$ satisfies the bodies of the first and third rule, yielding $T_{\mathcal{P}}(\mathcal{M}_1) = \{ q, r \}$ and thus

$$T_{\mathcal{K}}(\mathcal{M}_1) = [\![T_{\mathcal{P}}(\mathcal{M}_1) \cup \mathcal{O}]\!] = [\![\{ q, r, p \lor \neg q \}]\!] = [\![\{ p, q, r \}]\!] = \mathcal{M}_2 \enspace .$$

The interpretation $\mathcal{M}_2$ is a fixed point of $T_{\mathcal{K}}$ because it satisfies the bodies of the same rules as $\mathcal{M}_1$ and further applications of $T_{\mathcal{K}}$ have no effect on it. Moreover, $\mathcal{M}_2$ is also the unique MKNF model of $\mathcal{K}$.

The process demonstrated above works in general. To formally establish it, we first note that $T_{\mathcal{K}}$ is a monotonic function.

Proposition 3.4 (Monotonicity of $T_{\mathcal{K}}$)**.** *Let $\mathcal{K}$ be a positive MKNF knowledge base. Then $T_{\mathcal{K}}$ is a monotonic function on the complete lattice $(\mathfrak{M}, \subseteq)$.*

Proof. See Appendix B, page 189. □

[1] Note that usually it is the *least* element that is considered in analogical constructions from Logic Programming and the *least fixed point* of an operator is computed. The same could be done here but we refrain from doing so as it requires one to use a partial order on $\mathfrak{M}$ that is the reverse of the usual subset relation and this may easily cause confusion.

[2] To make this demonstration simpler, we assume that the ontology $\mathcal{O}$ is a set of propositional formulae and $\kappa(\mathcal{O}) = \mathcal{O}$.

Hence, by the Knaster-Tarski theorem, $T_{\mathcal{K}}$ has the greatest fixed point. This can be either the empty set, in which case $\mathcal{K}$ has no MKNF model, or it coincides with the unique MKNF model of $\mathcal{K}$. In other words, $T_{\mathcal{K}}$ offers a constructive characterisation of the semantics of positive MKNF knowledge bases.

Proposition 3.5 (MKNF Model of a Positive MKNF Knowledge Base)**.** *Let $\mathcal{K}$ be a positive MKNF knowledge base. An MKNF interpretation is an MKNF model of $\mathcal{K}$ if and only if it is the greatest fixed point of $T_{\mathcal{K}}$.*

Proof. See Appendix B, page 190. □

To add support for default negation in bodies of MKNF rules, we can use essentially the same strategy as the one used for defining stable models (Gelfond and Lifschitz, 1988). Given an MKNF knowledge base $\mathcal{K}$, we first eliminate default negation by forming the reduct of $\mathcal{K}$ w.r.t. a candidate model $\mathcal{M}$. This reduct consists of the original ontology and positive parts of rules with negative bodies satisfied in $\mathcal{M}$.

Definition 3.6 (Reduct of an MKNF Knowledge Base)**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base and $\mathcal{M}$ an MKNF interpretation. The *reduct of $\mathcal{K}$ w.r.t. $\mathcal{M}$* is the MKNF knowledge base $\mathcal{K}^{\mathcal{M}} = \left(\mathcal{O}, \mathcal{P}^{\mathcal{M}}\right)$ where

$$\mathcal{P}^{\mathcal{M}} = \left\{ H(\pi) \leftarrow B(\pi)^{+}. \;\middle|\; \pi \in \mathcal{P} \land \mathcal{M} \models \kappa\left(\sim B(\pi)^{-}\right) \right\} \quad .$$

The following example shows that using the reduct we can determine the semantics of an MKNF knowledge base $\mathcal{K}$: an interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M}$ is the MKNF model of the positive MKNF knowledge base $\mathcal{K}^{\mathcal{M}}$.

Example 3.7 (Determining MKNF Models Using the Reduct)**.** Let $\mathcal{K}' = (\mathcal{O}, \mathcal{P}')$ be an MKNF knowledge base where $\mathcal{O} = \{ p \lor \neg q \}$ is as in Example 3.3 and

$$\begin{aligned} \mathcal{P}' : \quad & q \leftarrow r. \\ & q \leftarrow s. \\ & r \leftarrow \sim s. \\ & s \leftarrow \sim r. \end{aligned}$$

The reduct of $\mathcal{K}'$ w.r.t. $\mathcal{M}_2 = [\![\{ p, q, r \}]\!]$ is $(\mathcal{K}')^{\mathcal{M}_2} = \mathcal{K} = (\mathcal{O}, \mathcal{P})$ where $\mathcal{P}$ is as in Example 3.3. This is because $\mathcal{M}_2$ does not satisfy the negative body $\sim r$ of the last rule in $\mathcal{P}'$, so the rule is discarded, while it satisfies the negative body $\sim s$ of the third rule, so the rule turns into the fact $(r.)$ in the reduct. We have already shown that $\mathcal{M}_2$ is the MKNF model of $\mathcal{K}$. It is also one of the two MKNF models of $\mathcal{K}'$. The other MKNF model, $\mathcal{M}_2' = [\![\{ p, q, s \}]\!]$ can be obtained in a similar manner, by first forming the reduct of $\mathcal{K}'$ w.r.t. $\mathcal{M}_2'$ and then verifying that $\mathcal{M}_2'$ is the MKNF model of the reduct.

As the following proposition shows, this relationship holds in general.

Proposition 3.8 (MKNF Model of an MKNF Knowledge Base)**.** *Let $\mathcal{K}$ be an MKNF knowledge base. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if it is the MKNF model of $\mathcal{K}^{\mathcal{M}}$.*

Proof. See Appendix B, page 191. □

3.2 Updating Consequence Operator

We have seen in the previous section that MKNF models of an MKNF knowledge base $\mathcal{K}$ can be characterised in terms of reducts and the consequence operator $T_{\mathcal{K}}$, similarly as with stable models of a logic program. In order to deal with updates to the ontology part of the knowledge base, we can modify the consequence operator so that the ontology gets updated accordingly whenever the operator is applied. Nevertheless, this strategy does not let us perform rule updates as they are not performed on the level of models and instead rely on rule syntax, making it difficult to resolve conflicts between rules and ontology axioms. Therefore, we constrain ourselves to dealing with are DMKBs that have *static rules*, as defined here:

Definition 3.9 (DMKB with Static Rules)**.** We say that a DMKB $\langle(\mathcal{O}_i, \mathcal{P}_i)\rangle_{i<n}$ has *static rules* if $\mathcal{P}_i = \emptyset$ for all i such that $0 < i < n$.

So assume that we are given a DMKB with static rules $\boldsymbol{K}$ and that $\boldsymbol{K}$ is *positive*, i.e. all component MKNF knowledge bases of $\boldsymbol{K}$ are positive. Furthermore, suppose that we want to use the first-order update operator $\diamond$ to perform ontology updates. We can use $\diamond$ within the consequence operator to reflect the updates in $\boldsymbol{K}$ as follows:

Definition 3.10 (Updating Consequence Operator)**.** Let $\diamond$ be a first-order update operator and $\boldsymbol{K} = \langle(\mathcal{O}_i, \mathcal{P}_i)\rangle_{i<n}$ a positive DMKB with static rules. The *updating immediate consequence operator* $T^{\diamond}_{\boldsymbol{K}}$ is defined for all $\mathcal{M} \in \mathfrak{M}$ as follows:

$$T^{\diamond}_{\boldsymbol{K}}(\mathcal{M}) = [\![(T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0)) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \enspace .$$

The following example illustrates the workings of the introduced consequence operator when we adopt Winslett's operator to perform ontology updates.

Example 3.11 (Iterating the Updating Consequence Operator)**.** Consider the DMKB $\boldsymbol{K} = \langle \mathcal{K}_0, \mathcal{K}_1 \rangle$ where $\mathcal{K}_0 = (\mathcal{O}_0, \mathcal{P}_0)$, $\mathcal{K}_1 = (\mathcal{O}_1, \emptyset)$ and $\mathcal{O}_0$, $\mathcal{P}_0$ and $\mathcal{O}_1$ are as follows:

$$\mathcal{O}_0 : \; p \lor \neg q \qquad \mathcal{P}_0 : \; \begin{array}{l} q \leftarrow r. \\ q \leftarrow s. \\ r. \\ s. \end{array} \qquad \mathcal{O}_1 : \; \neg r \land \neg s$$

Clearly, $\boldsymbol{K}$ has static rules. We can thus iteratively apply the updating consequence operator $T^{\diamond_{\mathsf{W}}}_{\boldsymbol{K}}$, starting from the MKNF interpretation $\mathcal{M}_0 = \mathfrak{I}$, until we reach a fixed point. After the first application we obtain the following:

$$\begin{aligned} T^{\diamond_{\mathsf{W}}}_{\boldsymbol{K}}(\mathcal{M}_0) &= [\![(T_{\mathcal{P}_0}(\mathcal{M}_0) \cup \mathcal{O}_0) \diamond_{\mathsf{W}} \mathcal{O}_1]\!] \\ &= [\![\{ r, s, p \lor \neg q \} \diamond_{\mathsf{W}} \{ \neg r, \neg s \}]\!] \\ &= [\![\{ p \lor \neg q, \neg r, \neg s \}]\!] = \mathcal{M}_1 \enspace . \end{aligned}$$

Furthermore, $T^{\diamond_{\mathsf{W}}}_{\boldsymbol{K}}(\mathcal{M}_1) = \mathcal{M}_1$ because $\mathcal{M}_1$ triggers only the facts $(r.)$ and $(s.)$ in $\mathcal{P}_0$, just as $\mathcal{M}_0$ did. It is not difficult to verify that $\mathcal{M}_1$ is the greatest fixed point of $T^{\diamond_{\mathsf{W}}}_{\boldsymbol{K}}$ and following the analogy with the static case, we can declare $\mathcal{M}_1$ to be the *dynamic MKNF model of* $\boldsymbol{K}$. Note that $\mathcal{M}_1$ does not satisfy the two facts in $\mathcal{P}_0$ since they were overridden by the updating ontology.

If we can show that $T^{\diamond}_{\boldsymbol{K}}$ is monotonic, then it is guaranteed to have the greatest fixed point which we can use to *assign a semantics to any positive DMKB with static rules*. As it turns out, $T^{\diamond}_{\boldsymbol{K}}$ is monotonic if $\diamond$ satisfies the principle (FO8.2).

Proposition 3.12 (Monotonicity of $T^{\diamond}_{\boldsymbol{K}}$)**.** *Let $\diamond$ be a first-order update operator and $\boldsymbol{K}$ a positive DMKB with static rules. If $\diamond$ satisfies (FO8.2), then $T^{\diamond}_{\boldsymbol{K}}$ is a monotonic function on the complete lattice $(\mathfrak{M}, \subseteq)$.*

Proof. See Appendix B, page 191. □

So assuming that $\diamond$ satisfies (FO8.2), we can declare the greatest fixed point of $T^{\diamond}_{\boldsymbol{K}}$ as the semantics of $\boldsymbol{K}$. The following definition establishes the notion of a $\diamond$*-dynamic MKNF model*.

Definition 3.13 (Semantics for Positive DMKBs with Static Rules)**.** Let $\diamond$ be a first-order update operator that satisfies (FO8.2) and $\boldsymbol{K}$ a positive DMKB with static rules. An MKNF interpretation $\mathcal{M}$ is a $\diamond$*-dynamic MKNF model of* $\boldsymbol{K}$ if it is the greatest fixed point of $T^{\diamond}_{\boldsymbol{K}}$;

Note that every positive DMKB with static rules has at most one $\diamond$-dynamic MKNF model. It may have no such model when the greatest fixed point of $T^{\diamond}_{\boldsymbol{K}}$ is the empty set because the empty set is not an MKNF interpretation.

Default negation can now be treated the same way as in the static case. We first establish the notion of a *reduct of a DMKB* in the expected way.

Definition 3.14 (Reduct of a DMKB)**.** Let $\boldsymbol{K} = \langle \mathcal{K}_i \rangle_{i<n}$ be a DMKB with static rules and $\mathcal{M}$ an MKNF interpretation. The *reduct of* $\boldsymbol{K}$ *w.r.t.* $\mathcal{M}$ is the DMKB $\boldsymbol{K}^{\mathcal{M}} = \langle \mathcal{K}_i^{\mathcal{M}} \rangle_{i<n}$.

In the following example we illustrate how the reduct can be used to determine the semantics of a DMKB with static rules.

Example 3.15 (Assigning Dynamic MKNF Models Using the Reduct)**.** Take the DMKB $\boldsymbol{K}' = \langle \mathcal{K}'_0, \mathcal{K}_1 \rangle$ where $\mathcal{K}'_0 = (\mathcal{O}, \mathcal{P}')$, $\mathcal{O}$ and $\mathcal{P}'$ are as in Example 3.7 and $\mathcal{K}_1$ is as in Example 3.11. Consider the MKNF interpretation $\mathcal{M}_1 = [\![\{ p \vee \neg q, \neg r, \neg s \}]\!]$. The reduct of $\boldsymbol{K}'$ w.r.t. $\mathcal{M}_1$ is $(\boldsymbol{K}')^{\mathcal{M}_1} = \boldsymbol{K}$ where $\boldsymbol{K}$ is the positive DMKB from Example 3.11. This is because because $\mathcal{M}_1$ satisfies both $\sim r$ and $\sim s$, so both rules with negative bodies from $\mathcal{P}'$ become facts in the reduct. Since $\mathcal{M}_1$ is also the $\diamond_{\mathsf{W}}$-dynamic MKNF model of $\boldsymbol{K}$, we declare it as the $\diamond_{\mathsf{W}}$*-dynamic MKNF model of* $\boldsymbol{K}'$.

Furthermore, note that $\mathcal{M}_1$ is the only MKNF interpretation satisfying this condition. While the initial MKNF knowledge base $\mathcal{K}'$ had two MKNF models, after the update it only has one because the generating cycle in $\mathcal{P}'$ was overridden by information in $\mathcal{K}_1$.

We can now define the dynamic MKNF models of an arbitrary DMKB with static rules $\boldsymbol{K}$ as those MKNF interpretations $\mathcal{M}$ that are MKNF models of the reduct $\boldsymbol{K}^{\mathcal{M}}$.

Definition 3.16 (Semantics for DMKBs with Static Rules)**.** Let $\diamond$ be a first-order update operator that satisfies (FO8.2) and $\boldsymbol{K}$ a DMKB with static rules. An MKNF interpretation $\mathcal{M}$ is a $\diamond$*-dynamic MKNF model of* $\boldsymbol{K}$ if it is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}^{\mathcal{M}}$.

Finally, we define a concrete consequence relation by considering the skeptical consequences of $\diamond_{\mathsf{W}}$-dynamic MKNF models, i.e. in this definition we use Winslett's first-order operator for performing ontology updates.

Definition 3.17 (Consequence Relation)**.** Let $\boldsymbol{K}$ be a DMKB with static rules and $\mathcal{T}$ an MKNF theory. We say that $\boldsymbol{K}$ *MKNF entails* $\mathcal{T}$, denoted by $\boldsymbol{K} \models_{\mathsf{MKNF}} \mathcal{T}$, if $\mathcal{M} \models \mathcal{T}$ for every $\diamond_{\mathsf{W}}$-dynamic MKNF model $\mathcal{M}$ of $\boldsymbol{K}$.

Consequence relations for other update operators as well as credulous and other types of logical consequence can be obtained analogously.

Let us now demonstrate the defined update semantics on a simple example:

Example 3.18 (Updating a DMKB with Static Rules)**.** Consider the following initial MKNF knowledge base $\mathcal{K}_0 = (\mathcal{T}, \mathcal{P})$ where

$$\mathcal{T}: \quad A \equiv B \sqcup C \tag{3.1}$$
$$D \equiv \neg A \sqcap \exists R^{-}.A \tag{3.2}$$
$$\mathcal{P}: \quad \neg A(\mathbf{x}) \leftarrow \sim A(\mathbf{x}). \tag{3.3}$$
$$R(\mathbf{x}, \mathbf{y}) \leftarrow A(\mathbf{x}), E(\mathbf{y}), \sim E(\mathbf{x}). \tag{3.4}$$

The TBox assertion (3.1) together with rule (3.3) define the concept A as a union of concepts B and C and they make this concept interpreted under the closed world assumption instead of the open world assumption, i.e. whenever for some constant a we cannot conclude that $A(a)$ is true, the rule (3.3) infers $\neg A(a)$. Assertion (3.2) defines concept D as those members b of $\neg A$ for which there exists some a from A with $R(a, b)$. Rule (3.4) infers $R(a, b)$ whenever a is in A but not in E and b is in E. Note that the schema variables $\mathbf{x}$, $\mathbf{y}$ are used merely as substitutes for the set of all ground instances of the rules.

Due to the issues that Winslett's operator has with TBox updates (c.f. Example 1.8), we consider only ABox updates and keep the TBox static throughout the process by reasserting it in each update. In Chapter 5 we show that the problems with TBox updates are not specific to Winslett's operator but extend to a large class of model-based update operators.

We also assume that the two constant symbols a, b are part of the language.

Initially, we can conclude by rule (3.3) that $\neg A(a)$ and $\neg A(b)$ are entailed by $\mathcal{K}_0$. Suppose we want to update $\mathcal{K}_0$ by the ABox $\mathcal{A}_1 = \{ A(a) \}$. We form the update $\mathcal{K}_1 = (\mathcal{A}_1 \cup \mathcal{T}, \emptyset)$ and the DMKB $\boldsymbol{K}_1 = \langle \mathcal{K}_0, \mathcal{K}_1 \rangle$ and conclude that

$$\boldsymbol{K}_1 \models_{\mathsf{MKNF}} \{ A(a), \neg A(b) \} \quad .$$

A further update by $\mathcal{A}_2 = \{ \neg B(a) \}$ introduces a possibility of $A(a)$ not being true in case $B(a)$ was true before and $C(a)$ was false. This renders the truth of $A(a)$ unknown and since A is interpreted under the closed world assumption, we can conclude that $A(a)$ is false. Formally, for $\mathcal{K}_2 = (\mathcal{A}_2 \cup \mathcal{T}, \emptyset)$ and $\boldsymbol{K}_2 = \langle \mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2 \rangle$ we can observe that

$$\boldsymbol{K}_2 \models_{\mathsf{MKNF}} \{ \neg A(a), \neg B(a), \neg A(b) \} \quad .$$

Now consider the ABox update $\mathcal{A}_3 = \{ C(a), E(b) \}$. Given (3.1), this reinstates $A(a)$. Furthermore, rule (3.4) can now infer $R(a, b)$ and by (3.2) we obtain $D(b)$. Formally, if we put $\mathcal{K}_3 = (\mathcal{A}_3 \cup \mathcal{T}, \emptyset)$ and $\boldsymbol{K}_3 = \langle \mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3 \rangle$, then it holds that

$$\boldsymbol{K}_3 \models_{\mathsf{MKNF}} \{ A(a), \neg B(a), C(a), \neg A(b), E(b), R(a, b), D(b) \} \quad .$$

In the next update $\mathcal{A}_4 = \{ E(a) \}$ we block the body of rule (3.4), which also prevents $D(b)$ from being inferred. Thus for $\mathcal{K}_4 = (\mathcal{A}_4 \cup \mathcal{T}, \emptyset)$ and $\boldsymbol{K}_4 = \langle \mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4 \rangle$ we have

$$\boldsymbol{K}_4 \models_{\mathsf{MKNF}} \{ A(a), \neg B(a), C(a), \neg A(b), E(b), E(a) \} \quad .$$

The last update $\mathcal{A}_5 = \{ \neg E(a), \neg R(a, b) \}$ illustrates how the conclusion of a rule may be overridden by an update – though the body of rule (3.4) is satisfied, its head does not become true since it is in direct conflict with $\mathcal{A}_5$. So if $\mathcal{K}_5 = (\mathcal{A}_5 \cup \mathcal{T}, \emptyset)$ and $\boldsymbol{K}_5 = \langle \boldsymbol{K}_0, \boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{K}_3, \boldsymbol{K}_4, \boldsymbol{K}_5 \rangle$, then

$$\boldsymbol{K}_5 \models_{\mathsf{MKNF}} \{ A(a), \neg B(a), C(a), \neg A(b), E(b), \neg E(a), \neg R(a,b) \} \enspace .$$

3.3 Properties and Relations

We now look at some basic formal properties of the introduced update framework for DMKBs with static rules. Throughout this section we assume that $\diamond$ is some first-order update operator that satisfies (FO8.2).

First we establish that our update semantics is faithful to the main ingredients it is based upon: the semantics of MKNF knowledge bases and the first-order update operator $\diamond$. The former property can be formulated as follows:

Theorem 3.19 (Faithfulness w.r.t. MKNF Knowledge Bases)**.** *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\langle \mathcal{K} \rangle$.*

Proof. See Appendix B, page 192. □

Note that a consequence of this result and of Propositions 2.17 and 2.19 is that the introduced update semantics is also faithful w.r.t. ontologies and stable models.

Turning to the relation with the first-order update operator $\diamond$, we show that if the initial program is empty, then the assigned dynamic MKNF model coincides with the semantics of updating the initial ontology with all subsequent ones in the DMKB. Formally:

Theorem 3.20 (Faithfulness w.r.t. First-Order Update Operator)**.** *Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \emptyset) \rangle_{i<n}$ be a DMKB. An MKNF interpretation $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M} = [\![\diamond \langle \mathcal{O}_i \rangle_{i<n}]\!]$.*

Proof. See Appendix B, page 192. □

Now we consider other important properties that are typically expected of an update semantics. The first one, known as the *principle of primacy of new information* (Dalal, 1988), guarantees that every dynamic MKNF model satisfies the most recent update. This can also be seen as a counterpart of the belief update postulate (FO1) and in order for the property to hold, the first-order update operator must satisfy (FO1).

Theorem 3.21 (Primacy of New Information)**.** *Suppose that $\diamond$ satisfies (FO1) and let $\boldsymbol{K} = \langle \mathcal{K}_i \rangle_{i<n}$ be a DMKB with static rules such that $n > 0$. If $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}$, then $\mathcal{M} \models \kappa(\mathcal{K}_{n-1})$.*

Proof. See Appendix B, page 192. □

The second property, inherited from the first-order update operator, states that updates by tautological ontologies do not influence the resulting models. This can be seen as a counterpart of the postulate (FO2.⊤) and is satisfied if the first-order update operator satisfies (FO2.⊤) and (FO4). Note that a similar property is violated by most existing rule update semantics (c.f. Section 2.8).

Theorem 3.22 (Immunity to Tautological Updates)**.** *Suppose that $\diamond$ satisfies (FO2.⊤) and (FO4) and let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$ be a DMKB with static rules such that $\mathcal{O}_j \equiv \emptyset$ for some j with $0 < j < n$ and*

$$\boldsymbol{K}' = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n \wedge i \neq j} \enspace .$$

Then $\boldsymbol{K}$ and $\boldsymbol{K}'$ have the same $\diamond$-dynamic MKNF models.

Proof. See Appendix B, page 192. □

The final property guarantees that our update semantics does not depend on the syntax of ontologies, only on their semantics. It essentially shows that substituting an ontology for an equivalent one at any point in the DMKB always leads to the same result. This property can be seen as a counterpart of postulate (FO4.2) and partially also postulate (FO4.1). It holds if the first-order update operator satisfies (FO4).

Theorem 3.23 (Syntax Independence)**.** *Suppose that $\diamond$ satisfies (FO4). Let $\boldsymbol{K} = \langle(\mathcal{O}_i, \mathcal{P}_i)\rangle_{i<n}$ and $\boldsymbol{K}' = \langle(\mathcal{O}'_i, \mathcal{P}'_i)\rangle_{i<n}$ be DMKBs with static rules such that $\mathcal{P}_0 = \mathcal{P}'_0$ and $\mathcal{O}_i \equiv \mathcal{O}'_i$ for all $i < n$. Then $\boldsymbol{K}$ and $\boldsymbol{K}'$ have the same $\diamond$-dynamic MKNF models.*

Proof. See Appendix B, page 193. □

Note that a similar property does not hold for the initial program $\mathcal{P}_0$. For example, programs such as

$$\mathcal{P}: \quad p. \quad q. \qquad \text{and} \qquad \mathcal{Q}: \quad p \leftarrow q. \quad q.$$

have the same stable models and are even strongly equivalent (Lifschitz et al., 2001), but an update by $\mathcal{O} = \neg q$ produces different results for $\mathcal{P}$ and $\mathcal{Q}$, respectively. More formally, $\langle(\emptyset, \mathcal{P}), (\mathcal{O}, \emptyset)\rangle$ has a $\diamond_{\mathsf{W}}$-dynamic MKNF model $\mathcal{M}$ such that $\mathcal{M} \models p$ while $\langle(\emptyset, \mathcal{Q}), (\mathcal{O}, \emptyset)\rangle$ has a $\diamond_{\mathsf{W}}$-dynamic MKNF model $\mathcal{M}'$ such that $\mathcal{M}' \not\models p$. We believe that this is in accord with intuitions regarding the two initial programs. It may be the case that for stronger notions of program equivalence that are better suited for updates, such as *update equivalence* proposed in (Leite, 2003), this property is satisfied. This is to some extent related with the developments in Chapters 7 and 8 where we investigate stronger notions of program equivalence and their suitability in the context of rule updates.

For similar reasons, it is not possible to prove properties that correspond to other belief update postulates. Consider for instance postulate (FO2). The first issue is that by relying on the defined consequence relation, we can only *approximate* its formulation as follows:

$$\text{If } (\mathcal{O}, \mathcal{P}) \models_{\mathsf{MKNF}} \mathcal{O}', \text{ then } \langle(\mathcal{O}, \mathcal{P}), (\mathcal{O}', \emptyset)\rangle \models_{\mathsf{MKNF}} (\mathcal{O}, \mathcal{P}).$$

In other words, instead of equivalence on the right-hand side of the postulate, we only have entailment. But even this weaker principle is not satisfied. As a counterexample, consider the program

$$\begin{array}{llll} \mathcal{P}: & p \leftarrow \sim q. & r \leftarrow p. & s \leftarrow p. \\ & q \leftarrow \sim p. & r \leftarrow q, \sim r. & \neg s \leftarrow q. \end{array}$$

and the ontology $\mathcal{O}' = \{ r, s \}$. Though it is true that $(\emptyset, \mathcal{P}) \models_{\mathsf{MKNF}} \mathcal{O}'$ because $(\emptyset, \mathcal{P})$ has a single MKNF model that entails both r and s, it is not true that $\langle(\emptyset, \mathcal{P}), (\mathcal{O}', \emptyset)\rangle \models_{\mathsf{MKNF}} (\emptyset, \mathcal{P})$ because $\langle(\emptyset, \mathcal{P}), (\mathcal{O}', \emptyset)\rangle$ has two $\diamond_{\mathsf{W}}$-dynamic MKNF models, one of which entails both q and s, so it does not satisfy the last rule in $\mathcal{P}$.

In fact, this behaviour is inherited from stable models which do not satisfy the very similar property of *cumulativity* (Makinson, 1988; Dix, 1995a). Hence it is likely that (FO2) is not going to be satisfied by any hybrid update semantics that is faithful to the stable models semantics.

A similar analysis can be also done for (FO3) if we identify "consistency" with existence of a ($\diamond$-dynamic) MKNF model. Thus the corresponding property would read as follows:

If both $(\mathcal{O}, \mathcal{P})$ and $(\mathcal{O}', \emptyset)$ have an MKNF model,
then $\langle (\mathcal{O}, \mathcal{P}), (\mathcal{O}', \emptyset) \rangle$ has a $\diamond$-dynamic MKNF model.

It is not difficult to show that this property is not satisfied – as a counterexample, consider the program $\mathcal{P} = \{ p \leftarrow q, \sim p. \}$ and ontology $\mathcal{O}' = \{ q \}$. Though both $(\emptyset, \mathcal{P})$ and $(\mathcal{O}', \emptyset)$ have MKNF models, the DMKB $\langle (\emptyset, \mathcal{P}), (\mathcal{O}', \emptyset) \rangle$ does not have a $\diamond_{\mathrm{W}}$-dynamic MKNF model.

The cases of most other belief update postulates are even more involved because they require notions such as disjunction of two MKNF knowledge bases and it is not clear how these can be defined appropriately.

3.4 Discussion

In this chapter we have proposed the first update semantics for MKNF knowledge bases, parametrised by a first-order update operator. It can essentially deal with a constrained but interesting scenario in which the rules represent static knowledge, policies, norms and default preferences, and the evolving ontology represents the open and dynamic environment. It can be used in realistic scenarios where the general concepts and rules are relatively fixed, and individuals tend to change their state frequently. This is the case of many real life institutions where data about stakeholders changes on a regular basis while the general rules and structures change only occasionally.

We have also demonstrated how the resulting semantics behaves if we pick Winslett's operator as the underlying update operator. Due to the problems that Winslett's operator has with performing TBox updates, in the example we constrain ourselves to ABox updates. In Chapter 5 we revisit this issue, showing that these problems are not specific to Winslett's operator but extend to all relevant model-based update operators. Hence TBox updates need to be studied further and unified with ABox updates before a suitable universal hybrid update semantics can be defined.

Subsequently, we turned to the properties of the proposed update semantics and proved that it is faithful to both the semantics for MKNF knowledge bases as well as to the underlying first-order update operator. We also looked more closely at other generic properties of the semantics, showing that it satisfies the principle of primacy of new information, is immune to tautological updates, and also syntax-independent w.r.t. the initial ontology an its updates. These properties roughly correspond to the belief update postulates (FO1), (FO2.$\top$) and (FO4).

On the other hand, the failure of our semantics to satisfy postulates such as (FO2) or (FO3) is not surprising. A wide range of classical update and revision postulates was already studied in the context of rule updates, only to find that most of them were not satisfied by existing rule update semantics (Eiter et al., 2002). Part of the problem is the fact that it is not clear how the original postulates should reinterpreted in the face of non-monotonic semantics such as stable or MKNF models.

4

Layered Dynamic MKNF Knowledge Bases

JOÃO: So you say that a general update semantics for hybrid knowledge bases is a mystery. How about particular scenarios? Can we deal with those?

MARTIN: Many use cases are simply rules on top of the ontology. The more complex ones, like Terry's scenario... it still seems that the ontology is not arbitrarily mixed with rules. It's more like there are some sort of ontology and rule layers that only share information to a certain extent.

JOÃO: So we could try to extract these layers and deal with them separately?

MARTIN: Perhaps...

JANUARY 2011, LISBON, PORTUGAL

We have seen in the previous chapter how to define a plausible update semantics for DMKBs with static rules. Despite the importance of this first step towards a universal hybrid update semantics, its applicability is limited since typically all parts of a knowledge base are subject to change. As an example, consider the following scenario where both an ontology and rules are needed to assess the risk of imported cargo.

Example 4.1 (MKNF Knowledge Base for Cargo Imports)**.** The customs service for any developed country assesses imported cargo for a variety of risk factors including terrorism, narcotics, food and consumer safety, pest infestation, tariff violations, and intellectual property rights.[1] Assessing this risk, even at a preliminary level, involves

[1] The system described here is not intended to reflect the policies of any country or agency.

extensive knowledge about commodities, business entities, trade patterns, government policies and trade agreements. Some of this knowledge may be external to a given customs agency: for instance the broad classification of commodities according to the international Harmonized Tariff System (HTS), or international trade agreements. Other knowledge may be internal to a customs agency, such as lists of suspected violators or of importers who have a history of good compliance with regulations. While some of this knowledge is relatively stable, much of it changes rapidly. Changes are made not only at a specific level, such as knowledge about the expected arrival date of a shipment; but at a more general level as well. For instance, while the broad HTS code for tomatoes (0702) does not change, the full classification and tariffs for cherry tomatoes for import into the US changes seasonally.

Figure 4.1 shows a simplified fragment $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ of such a knowledge base. In this fragment, a shipment has several attributes: the country of its origination, the commodity it contains, its importer and producer. The ontology contains a geographic classification, along with information about producers who are located in various countries. It also contains a classification of commodities based on their harmonised tariff information (HTS chapters, headings and codes, c.f. `http://www.usitc.gov/tata/hts`). Tariff information is also present, based on the classification of commodities. Finally, the ontology contains (partial) information about three shipments: s_1, s_2 and s_3. There is also a set of rules indicating information about importers, and about whether to inspect a shipment either to check for compliance of tariff information or for food safety issues.

In the present chapter we define a hybrid update semantics that can deal with scenarios such as the one described above. The semantics is parametrised by a first-order update operator and a rule update semantics. It can deal with an interesting class of DMKBs in which the interaction between the ontology and rules is limited but, unlike in Chapter 3, both the ontology and rules can be updated. One way to look at it is as a *modular combination* of a first-order update operator with a rule update semantics.

The main ideas for identifying this class of DMKBs come from the splitting theorems for Logic Programs (Lifschitz and Turner, 1994). We extract the essence of this work to characterise splittings of MKNF knowledge bases, sequences of ontologies, DLPs and DMKBs. We subsequently show that many semantics, such as MKNF models of MKNF knowledge bases, the models assigned by Winslett's first-order operator to a sequence of ontologies, or models assigned to DLPs by a number of rule update semantics, satisfy the splitting properties in essentially the same way as was shown by Lifschitz and Turner (1994) for stable models of logic programs.

Ultimately, these results enable us to define an update semantics for those DMKBs that can be split into a sequence of *ontology and rule layers* that share information using a rule-based interface. Each ontology layer is then updated using a first-order update operator, each rule layer using a rule update semantics, and the partial results are combined to obtain an overall dynamic MKNF model. We show that the defined hybrid update semantics enjoys several desirable characteristics, namely that it

- is faithful to the (static) semantics of MKNF knowledge bases;
- is faithful to the first-order update operator that it is based on;
- is faithful to the rule update semantics that it is based on;
- is in line with the hybrid update semantics introduced in Chapter 3;
- adheres to the principle of primacy of new information;
- properly deals with non-trivial updates in the scenario from Example 4.1.

$* * * \mathcal{O} * * *$

$\mathsf{Commodity} \equiv (\exists \mathsf{HTSCode}.\top)$

$\mathsf{CherryTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020020'}\})$

$\mathsf{GrapeTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020010'}\})$

$\mathsf{CherryTomato} \sqsubseteq \mathsf{Tomato}$

$\mathsf{CherryTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$0\})$

$\mathsf{GrapeTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$40\})$

$\mathsf{CherryTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$50\})$

$\mathsf{GrapeTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$100\})$

$\mathsf{EURegisteredProducer} \equiv (\exists \mathsf{RegisteredProducer}.\mathsf{EUCountry})$

$\mathsf{LowRiskEUCommodity} \equiv (\exists \mathsf{ExpeditableImporter}.\top) \sqcap (\exists \mathsf{CommodCountry}.\mathsf{EUCountry})$

$\mathsf{EdibleVegetable} \equiv (\exists \mathsf{HTSChapter}.\{\text{'07'}\})$

$\mathsf{Tomato} \equiv (\exists \mathsf{HTSHeading}.\{\text{'0702'}\})$

$\mathsf{Tomato} \sqsubseteq \mathsf{EdibleVegetable}$

$\mathsf{GrapeTomato} \sqsubseteq \mathsf{Tomato}$

$\mathsf{CherryTomato} \sqcap \mathsf{GrapeTomato} \sqsubseteq \bot$

$\mathsf{Bulk} \sqcap \mathsf{Prepackaged} \sqsubseteq \bot$

$\mathsf{ShpmtCommod}(s_1, c_1)$

$\mathsf{ShpmtImporter}(s_1, i_1)$

$\mathsf{ShpmtCommod}(s_2, c_2)$

$\mathsf{ShpmtImporter}(s_2, i_2)$

$\mathsf{ShpmtCountry}(s_2, portugal)$

$\mathsf{ShpmtCommod}(s_3, c_3)$

$\mathsf{ShpmtImporter}(s_3, i_3)$

$\mathsf{ShpmtCountry}(s_3, portugal)$

$\mathsf{RegisteredProducer}(p_1, portugal)$

$\mathsf{RegisteredProducer}(p_2, slovakia)$

$\mathsf{ShpmtDeclHTSCode}(s_1, \text{'07020020'})$

$\mathsf{CherryTomato}(c_1) \quad \mathsf{Bulk}(c_1)$

$\mathsf{ShpmtDeclHTSCode}(s_2, \text{'07020020'})$

$\mathsf{CherryTomato}(c_2) \quad \mathsf{Prepackaged}(c_2)$

$\mathsf{ShpmtDeclHTSCode}(s_3, \text{'07020010'})$

$\mathsf{GrapeTomato}(c_3) \quad \mathsf{Bulk}(c_3)$

$\mathsf{ShpmtProducer}(s_3, p_1)$

$\mathsf{EUCountry}(portugal)$

$\mathsf{EUCountry}(slovakia)$

$* * * \mathcal{P} * * *$

$\mathsf{AdmissibleImporter}(\mathbf{x}) \leftarrow \sim\mathsf{SuspectedBadGuy}(\mathbf{x}).$

$\mathsf{SuspectedBadGuy}(i_1).$

$\mathsf{ApprovedImporterOf}(i_2, \mathbf{x}) \leftarrow \mathsf{EdibleVegetable}(\mathbf{x}).$

$\mathsf{ApprovedImporterOf}(i_3, \mathbf{x}) \leftarrow \mathsf{GrapeTomato}(\mathbf{x}).$

$\mathsf{CommodCountry}(\mathbf{x}, \mathbf{y}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{z}, \mathbf{x}), \mathsf{ShpmtCountry}(\mathbf{z}, \mathbf{y}).$

$\mathsf{ExpeditableImporter}(\mathbf{x}, \mathbf{y}) \leftarrow \mathsf{AdmissibleImporter}(\mathbf{y}), \mathsf{ApprovedImporterOf}(\mathbf{y}, \mathbf{x}).$

$\mathsf{CompliantShpmt}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{HTSCode}(\mathbf{y}, \mathbf{z}), \mathsf{ShpmtDeclHTSCode}(\mathbf{x}, \mathbf{z}).$

$\mathsf{RandomInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{Random}(\mathbf{y}).$

$\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{RandomInspection}(\mathbf{x}).$

$\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \sim\mathsf{LowRiskEUCommodity}(\mathbf{y}).$

$\mathsf{FullInspection}(\mathbf{x}) \leftarrow \sim\mathsf{CompliantShpmt}(\mathbf{x}).$

$\mathsf{FullInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{Tomato}(\mathbf{y}), \mathsf{ShpmtCountry}(\mathbf{x}, slovakia).$

Figure 4.1: MKNF knowledge base for Cargo Imports

In order to use a rule update semantics for updating MKNF rules, we constrain ourselves to a generalised atom base that consists of objective literals, meaning that MKNF programs coincide with logic programs. For the same reason we assume that every rule is ground and has exactly one literal in its head. Furthermore, in this chapter we do not consider the equality predicate because it interferes with language conservation of Winslett's first-order update operator (c.f. Example 2.35). In the rest of this chapter we implicitly work under these assumptions.

The remainder of this chapter is structured as follows: Section 4.1 introduces an abstraction of splitting properties that is applicable to many logical frameworks and in Section 4.2 we use this abstraction to show that the semantics of MKNF knowledge bases as well as many update semantics satisfy the splitting properties. In Section 4.3 we use the previously established results to identify a class of layered DMKBs to which we assign an update semantics by modularly combining first-order updates with rule updates. Section 4.4 is concerned with formal properties of the resulting hybrid update semantics and illustrates how it can be applied to the scenario described in Example 4.1. We discuss the future directions in Section 4.5.

The relevant proofs can be found in Appendix C. A preliminary version of this work has been published in (Slota et al., 2011). The present contribution additionally contains a formulation of the generic splitting properties and is more general since it allows for various combinations of first-order and rule update semantics, not just for one fixed pair as in (Slota et al., 2011).

4.1 Generalised Splitting Properties

The splitting properties were first studied by Lifschitz and Turner (1994) in the context of Logic Programs, generalising the notion of *program stratification*. Roughly speaking, the idea is to define a condition under which the stable models of a program $\mathcal{P}$ can be completely determined from the stable models of its subprograms. This is certainly true if the subprograms are constructed over mutually disjoint sets of objective literals – indeed, in this case every stable model of $\mathcal{P}$ is a union of stable models of its subprograms. The same holds vice versa if we check for consistency, i.e. every consistent union of stable models of subprograms is a stable model of $\mathcal{P}$.

The splitting properties take this idea further by allowing subprograms to share literals in a constrained, cascading manner. Assuming that we aim for a splitting into two subprograms, one of them can "feed" information into the second one. The subprograms are then called the *bottom* and *top* of $\mathcal{P}$ and the condition imposed on them is that literals shared between them must not occur in heads of rules in the top. This essentially ensures that rules in the top cannot influence inferences made in the bottom. It follows that each stable model of $\mathcal{P}$ is a union of a stable model X of the bottom and of a stable model Y of the top in which all shared atoms have been pre-interpreted under X. The converse holds as well if consistency of $X \cup Y$ is ensured. As a matter of fact, the same relationship holds if we split $\mathcal{P}$ into an arbitrary sequence of layers where each layer is allowed to "feed" information into the following ones.

Our aim in this chapter is to define an update semantics for DMKBs that consist of one or more ontology and rule layers that may feed information into subsequent layers. Updates of each layer, depending on whether it is an ontology or a rule layer, are handled by a first-order update operator or by a rule update semantics, respectively. The resulting models are then collected and an overall dynamic MKNF model is assembled. These ideas are materialised in Section 4.3 where we also show that if both the first-order update operator and rule update semantics have the *splitting property*, then regardless of which splitting of the DMKB we pick, we arrive at the same set of dynamic MKNF models.

Since we rely so heavily on splitting properties, we first give their generic formulation that can be instantiated for a particular formalism – be it default logic, as was done in (Turner, 1996), or MKNF knowledge bases, first-order update operators, rule update and hybrid update semantics, as we do in subsequent sections of this chapter. The abstractions we use for this purpose are inspired by similar abstractions in (Brewka and Eiter, 2007).

We consider logical formalisms and their semantics from an abstract perspective, as established by the following definition:

Definition 4.2 (Logical Formalism and Semantics)**.** A *logical formalism* is a pair $(\mathfrak{T}, \mathfrak{S})$ where $\mathfrak{T}$ denotes a set of syntactically correct theories and $\mathfrak{S}$ a set of semantic structures to be assigned to such theories. A *semantics* S *for* $(\mathfrak{T}, \mathfrak{S})$ is given by a partial function $[\![\cdot]\!]_{\mathsf{S}} : \mathfrak{T} \to 2^{\mathfrak{S}}$ that assigns sets of acceptable belief states from $\mathfrak{S}$ to theories from $\mathfrak{T}$.

For the rest of this section we assume that the logical formalism $(\mathfrak{T}, \mathfrak{S})$ is fixed but the semantics S is not. We also assume that there exists a binary operation $\uplus$ such that for all semantic structures $\mathcal{X}, \mathcal{Y} \in \mathfrak{S}$, $\mathcal{X} \uplus \mathcal{Y}$ denotes a structure that combines information from $\mathcal{X}$ and $\mathcal{Y}$. Note that if $\mathcal{X}$ is inconsistent with $\mathcal{Y}$, then $\mathcal{X} \uplus \mathcal{Y}$ need not belong to $\mathfrak{S}$. For example, if $\mathfrak{S}$ is the set of ASP interpretations, then $I \uplus J = I \cup J$ and if $I \cup J$ contains a pair of complementary literals, then it is not itself an ASP interpretation. We also assume that $\uplus$ has a neutral element $\mathbf{0} \in \mathfrak{S}$, i.e. $\mathcal{X} \uplus \mathbf{0} = \mathbf{0} \uplus \mathcal{X} = \mathcal{X}$ for all $\mathcal{X} \in \mathfrak{S}$, and that $[\![\emptyset]\!]_{\mathsf{S}} = \{ \mathbf{0} \}$. In case of ASP interpretations this neutral element is $\emptyset$.

The *splitting problem for* $(\mathfrak{T}, \mathfrak{S})$ is specified by

- defining the *splitting sets for every theory* $\mathcal{T} \in \mathfrak{T}$ and
- defining, for every $\mathcal{T} \in \mathfrak{T}$, every splitting set U for $\mathcal{T}$ and every $\mathcal{X} \in \mathfrak{S}$, the theories $b_U(\mathcal{T})$, $t_U(\mathcal{T})$ and $e_U(\mathcal{T}, \mathcal{X})$.

Intuitively, a splitting set U for a theory $\mathcal{T}$ is a set of syntactic building blocks, such as literals or predicate symbols, such that $\mathcal{T}$ can be split in two parts: a part that defines the semantics of elements of U, and of no others, and a part that defines the semantics of the remaining elements based on the semantics of elements from U. The former is called the *bottom of* $\mathcal{T}$ *relative to* U and denoted by $b_U(\mathcal{T})$. The latter is the *top of* $\mathcal{T}$ *relative to* U and is denoted by $t_U(\mathcal{T})$. The set $e_U(\mathcal{T}, \mathcal{X})$ is the *reduct of* $\mathcal{T}$ *relative to* U and is obtained from the top $t_U(\mathcal{T})$ by pre-interpreting elements of U in $\mathcal{X}$. The following example illustrates these notions:

Example 4.3 (Splitting Set, Bottom, Top and Reduct of Logic Programs)**.** In case of logic programs under the stable models semantics, $\mathfrak{T}$ is the set of all logic programs, $\mathfrak{S}$ is the set of all ASP interpretations and every splitting set is some set of objective literals (Lifschitz and Turner, 1994). Consider the program

$$\begin{aligned} \mathcal{P} : \quad & p. \\ & q \leftarrow p, {\sim}r. \end{aligned}$$

One splitting set for this program is $U = \{ p \}$ because $\mathcal{P}$ can be split in two sets, $b_U(\mathcal{P}) = \{ p. \}$ and $t_U(\mathcal{P}) = \{ q \leftarrow p, {\sim}r. \}$, such that $b_U(\mathcal{P})$ only contains literals from U and $t_U(\mathcal{P})$ does not contain literals from U in heads of rules.

Furthermore, the reduct $e_U(\mathcal{P}, J)$ depends on the ASP interpretation J. If $J \models p$, then $e_U(\mathcal{P}, J) = \{ q \leftarrow {\sim}r. \}$ while if $J \not\models p$, then $e_U(\mathcal{P}, J) = \emptyset$. In other words, $e_U(\mathcal{P}, J)$ consists of rules from $t_U(\mathcal{P})$ with all literals from U pre-interpreted in J.

In the next section we formally define splitting sets, bottoms and reducts for various formalisms, such as MKNF knowledge bases, finite sequences of ontologies, DLPs and DMKBs. Nevertheless, the specifics of these definitions are not required for defining what it means for a semantics S to *satisfy the splitting properties*. Assuming that the notions of splitting set, bottom, top and reduct are known for the logical formalism $(\mathcal{T}, \mathcal{S})$, we can define a *solution w.r.t. a splitting set* as follows:

Definition 4.4 (Solution w.r.t. a Splitting Set)**.** Let S be a semantics for $(\mathcal{T}, \mathcal{S})$ and U a splitting set for a theory $\mathcal{T} \in \mathcal{T}$. An *S-solution to $\mathcal{T}$ w.r.t. U* is a pair of semantic structures $(\mathcal{X}, \mathcal{Y})$ such that $\mathcal{X} \in [\![b_U(\mathcal{T})]\!]_S$ and $\mathcal{Y} \in [\![e_U(\mathcal{T}, \mathcal{X})]\!]_S$ and $\mathcal{X} \uplus \mathcal{Y} \in \mathcal{S}$.

The splitting set property requires that the models assigned to a theory $\mathcal{T}$ correspond one to one with the solutions to $\mathcal{T}$ w.r.t. some splitting set. Formally:

Definition 4.5 (Splitting Set Property)**.** We say that a semantics S for $(\mathcal{T}, \mathcal{S})$ *satisfies the splitting set property* if for all theories $\mathcal{T} \in \mathcal{T}$ for which $[\![\mathcal{T}]\!]_S$ is defined and every splitting set U for $\mathcal{T}$,

$$[\![\mathcal{T}]\!]_S = \{\, \mathcal{X} \uplus \mathcal{Y} \mid (\mathcal{X}, \mathcal{Y}) \text{ is an S-solution to } \mathcal{T} \text{ w.r.t. } U \,\} \quad .$$

If, instead of a single splitting set, we consider a sequence of such sets, we can divide a theory into a sequence of layers and formulate a generalised version of the splitting set property. This part of the theory relies on transfinite sequences of sets, so we first introduce the following basic concepts:

Definition 4.6 (Sequence)**.** A *(transfinite) sequence* is a family whose index set is an initial segment of ordinals, $\{\, \alpha \mid \alpha < \mu \,\}$. The ordinal μ is the *length* of the sequence. A sequence of sets $\langle U_\alpha \rangle_{\alpha<\mu}$ is *monotone* if $U_\beta \subseteq U_\alpha$ whenever $\beta \leq \alpha$, and *continuous* if, for each limit ordinal $\alpha < \mu$, $U_\alpha = \bigcup_{\beta<\alpha} U_\beta$. A sequence $\langle U_i \rangle_{i<n}$ is *finite* if $n < \omega$.

Assuming that all splitting sets are subsets of some fixed set $\mathcal{U}$, we define a *splitting sequence* as follows:

Definition 4.7 (Splitting Sequence)**.** A *splitting sequence* for a theory $\mathcal{T} \in \mathcal{T}$ is a monotone, continuous sequence $\langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{T}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathcal{U}$.

In order to define a *solution w.r.t. a splitting sequence*, we need to collect models of layers of $\mathcal{T}$ induced by the splitting sequence. The first layer of $\mathcal{T}$ is the part of $\mathcal{T}$ that only describes elements from U_0. Formally, this is exactly $b_{U_0}(\mathcal{T})$, so we obtain $\mathcal{X}_0$ as one of its models. Proceeding inductively, for every ordinal $\alpha + 1 < \mu$, the corresponding layer of $\mathcal{T}$ is the part of $\mathcal{T}$ that describes elements from $U_{\alpha+1}$, with elements from U_α pre-interpreted in models of previous layers. Given our notation, and assuming that the binary operator $\uplus$ can be generalised to arbitrary subsets of $\mathcal{S}$, $\mathcal{X}_{\alpha+1}$ is chosen as one of the models of $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{T}), \biguplus_{\beta\leq\alpha} \mathcal{X}_\beta)$. Limit ordinals form a marginal case – since the splitting sequence is continuous, the set $U_\alpha \setminus \bigcup_{\beta<\alpha} U_\beta$ is empty for every limit ordinal α, hence the corresponding layer of $\mathcal{T}$ is empty as well and, consequently, $\mathcal{X}_\alpha = \mathbf{0}$. These observations lead to the following definition:

Definition 4.8 (Solution w.r.t. a Splitting Sequence)**.** Let S be a semantics for $(\mathcal{T}, \mathcal{S})$ and $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ a splitting sequence for a theory $\mathcal{T} \in \mathcal{T}$. An *S-solution to $\mathcal{T}$ w.r.t. $\boldsymbol{U}$* is a sequence of semantic structures $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ such that

1. $\mathcal{X}_0 \in [\![b_{U_0}(\mathcal{T})]\!]_S$;

2. For any ordinal α such that $\alpha + 1 < \mu$,

$$\mathcal{X}_{\alpha+1} \in \left[\!\!\left[e_{U_\alpha} \left(b_{U_{\alpha+1}}(\mathcal{T}), \biguplus_{\beta \leq \alpha} \mathcal{X}_\beta \right) \right]\!\!\right]_{\mathsf{S}} \quad ;$$

3. For any limit ordinal $\alpha < \mu$, $\mathcal{X}_\alpha = \mathbf{0}$;
4. $\biguplus_{\alpha < \mu} \mathcal{X}_\alpha \in \mathcal{S}$.

The splitting sequence property is now a straightforward adaptation of the splitting set property.

Definition 4.9 (Splitting Sequence Property)**.** We say that a semantics S for $(\mathfrak{T}, \mathcal{S})$ *satisfies the splitting sequence property* if for all theories $\mathcal{T} \in \mathfrak{T}$ for which $[\![\mathcal{T}]\!]_{\mathsf{S}}$ is defined and every splitting sequence $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha < \mu}$ for $\mathcal{T}$,

$$[\![\mathcal{T}]\!]_{\mathsf{S}} = \left\{ \biguplus_{\alpha < \mu} \mathcal{X}_\alpha \;\middle|\; \langle \mathcal{X}_\alpha \rangle_{\alpha < \mu} \text{ is an } \mathsf{S}\text{-solution to } \mathcal{T} \text{ w.r.t. } \boldsymbol{U} \right\} \quad .$$

4.2 Semantics with Splitting Properties

Now we take a look at instantiations of splitting properties for the cases of MKNF knowledge bases, ontology updates and rule updates. Unlike in (Lifschitz and Turner, 1994), we consider sets of predicate symbols instead of sets of ground literals as our splitting sets. By doing this, the set of ground literals with the same predicate symbol is considered either completely included in a splitting set or completely excluded from it. While this makes our approach less general than if we considered each ground literal individually, it simplifies considerably the splitting of TBoxes because they contain axioms with an implicit universal quantifier.

4.2.1 MKNF Knowledge Bases

We instantiate splitting properties for MKNF knowledge bases as follows:

- The set of theories $\mathfrak{T}$ is the set of all MKNF knowledge bases;
- The set of semantic structures $\mathcal{S}$ is the set of all MKNF interpretations;
- $\uplus$ is the set intersection $\cap$ with the neutral element $\mathbf{0} = \mathcal{I}$;
- The semantic function $[\![\cdot]\!]_{\mathsf{S}}$ returns all MKNF models of the argument MKNF knowledge base.

A splitting set for an MKNF knowledge base is defined analogically to a splitting set for a logic program, with the additional constraint that each ontology axiom must either use only predicate symbols from the splitting set, or only predicate symbols outside the splitting set.

Definition 4.10 (Splitting Set for an Ontology, Program and MKNF Knowledge Base)**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base and $U \subseteq \mathfrak{P}$ a set of predicate symbols. We say that U is a

- *splitting set for* $\mathcal{O}$ if for every axiom $\phi \in \mathcal{O}$, if $\mathsf{pr}(\phi) \cap U \neq \emptyset$, then $\mathsf{pr}(\phi) \subseteq U$;
- *splitting set for* $\mathcal{P}$ if for every rule $\pi \in \mathcal{P}$, if $\mathsf{pr}(H(\pi)) \cap U \neq \emptyset$, then $\mathsf{pr}(\pi) \subseteq U$;
- *splitting set for* $\mathcal{K}$ if it is a splitting set for both $\mathcal{O}$ and $\mathcal{P}$.

The bottom of an MKNF knowledge base relative to a splitting set U contains ontology axioms and rules that contain only predicate symbols from U. The top, on the other hand, contains the remaining ontology axioms and rules. Formally:

Definition 4.11 (Bottom and Top of an Ontology, Program and MKNF Knowledge Base)**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base and $U \subseteq \mathfrak{P}$ a set of predicate symbols. We define the *bottom of $\mathcal{O}$ and $\mathcal{P}$ relative to U* as

$$b_U(\mathcal{O}) = \{ \phi \in \mathcal{O} \mid \mathsf{pr}(\phi) \subseteq U \} \qquad \text{and} \qquad b_U(\mathcal{P}) = \{ \pi \in \mathcal{P} \mid \mathsf{pr}(\pi) \subseteq U \} \enspace .$$

The *bottom of $\mathcal{K}$ relative to U* is $b_U(\mathcal{K}) = (b_U(\mathcal{O}), b_U(\mathcal{P}))$.

The *top of $\mathcal{O}$, $\mathcal{P}$ and $\mathcal{K}$* is defined as $t_U(\mathcal{O}) = \mathcal{O} \setminus b_U(\mathcal{O})$, $t_U(\mathcal{P}) = \mathcal{P} \setminus b_U(\mathcal{P})$ and $t_U(\mathcal{K}) = (t_U(\mathcal{O}), t_U(\mathcal{P}))$, respectively.

Next, we need to define the reduct that makes it possible to use an MKNF model $\mathcal{X}$ of the bottom of $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ to simplify the top of $\mathcal{K}$. The top of the ontology $t_U(\mathcal{O})$ cannot be reduced in this manner because it only contains predicate symbols that do not belong to U. In case of the top of the program $t_U(\mathcal{P})$, we can discard rules that contain a body literal L with $\mathsf{pr}(L) \subseteq U$ that is not satisfied in $\mathcal{X}$, and eliminate the remaining literals L with $\mathsf{pr}(L) \subseteq U$. This is formally captured as follows:

Definition 4.12 (Reduct of a Program and MKNF Knowledge Base)**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base, $U \subseteq \mathfrak{P}$ a set of predicate symbols and $\mathcal{X} \in \mathfrak{M}$. We define the *reduct of $\mathcal{P}$ relative to U and $\mathcal{X}$* as

$$\begin{aligned} e_U(\mathcal{P}, \mathcal{X}) = \{ H(\pi) \leftarrow \{ L \in B(\pi) \mid \mathsf{pr}(L) \subseteq \mathfrak{P} \setminus U \}. \mid \pi \in t_U(\mathcal{P}) \\ \land \mathcal{X} \models \kappa(\{ L \in B(\pi) \mid \mathsf{pr}(L) \subseteq U \}) \} \enspace . \end{aligned}$$

The *reduct of $\mathcal{K}$ relative to U and $\mathcal{X}$* is $e_U(\mathcal{K}, \mathcal{X}) = (t_U(\mathcal{O}), e_U(\mathcal{P}, \mathcal{X}))$.

The definitions of splitting properties now follow from the generic ones defined in Section 4.1. As the following theorem shows, the MKNF models semantics satisfies both splitting properties:

Theorem 4.13 (Splitting Theorem for MKNF Knowledge Bases)**.** *The MKNF models semantics for MKNF knowledge bases satisfies the splitting set and splitting sequence properties.*

Proof. See Appendix C, page 206. □

An MKNF knowledge base can be split in a number of different ways. For example, $\emptyset$ and $\mathfrak{P}$ are splitting sets for any MKNF knowledge base and sequences such as $\langle \mathfrak{P} \rangle$, $\langle \emptyset, \mathfrak{P} \rangle$ are splitting sequences for any MKNF knowledge base. The following example shows a more elaborate splitting sequence for the Cargo Imports knowledge base from Example 4.1.

Example 4.14 (Splitting the Cargo Imports Knowledge Base)**.** Consider the MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ presented in Figure 4.1. One of the non-trivial splitting sequences for $\mathcal{K}$ is $\boldsymbol{U} = \langle U_0, U_1, U_2, U_3 \rangle$, where

$$
\begin{aligned}
U_0 = \quad & \{ \mathsf{Commodity}/1, \mathsf{EdibleVegetable}/1, \mathsf{Tomato}/1, \mathsf{CherryTomato}/1,\\
& \mathsf{GrapeTomato}/1, \mathsf{HTSCode}/2, \mathsf{HTSChapter}/2, \mathsf{HTSHeading}/2, \mathsf{Bulk}/1,\\
& \mathsf{Prepackaged}/1, \mathsf{TariffCharge}/2, \mathsf{ShpmtCommod}/2, \mathsf{ShpmtImporter}/2,\\
& \mathsf{ShpmtDeclHTSCode}/2, \mathsf{ShpmtProducer}/2, \mathsf{ShpmtCountry}/2 \} \enspace ,\\
U_1 = U_0 \cup & \{ \mathsf{AdmissibleImporter}/1, \mathsf{SuspectedBadGuy}/1, \mathsf{ApprovedImporterOf}/2 \} \enspace ,\\
U_2 = U_1 \cup & \{ \mathsf{RegisteredProducer}/2, \mathsf{EUCountry}/1, \mathsf{EURegisteredProducer}/1,\\
& \mathsf{CommodCountry}/2, \mathsf{ExpeditableImporter}/2, \mathsf{LowRiskEUCommodity}/1 \} \enspace ,\\
U_3 = U_2 \cup & \{ \mathsf{CompliantShpmt}/1, \mathsf{Random}/1, \mathsf{RandomInspection}/1, \mathsf{PartialInspection}/1,\\
& \mathsf{FullInspection}/1 \} \enspace .
\end{aligned}
$$

This splitting sequence divides $\mathcal{K}$ into the four layers shown in Figure 4.2. The first layer contains all ontological knowledge regarding commodity types as well as information about shipments. The second layer contains rules for classifying importers using internal records and information from the first layer. The third layer contains axioms with geographic classification, information about registered producers and, based on information about commodities and importers from the first two layers, it defines low risk commodities coming from the European Union. The final layer contains rules for deciding which shipments should be inspected based on information from previous layers.

4.2.2 Ontology Updates

As described in Section 2.7, we use first-order update operators to deal with updates of ontologies. In this context, given a first-order update operator $\diamond$, the generic splitting properties can be instantiated as follows:

- The set of theories $\mathfrak{T}$ contains finite sequences of first-order theories;
- The set of semantic structures $\mathcal{S} = 2^{\mathcal{I}}$ contains sets of interpretations that adopt the standard names assumption and interpret equality as a congruence relation (c.f. Section 2.4 for the definition of $\mathcal{I}$ and Section 2.6 for motivation on its use as a basis for first-order updates);
- Similarly as before, $\uplus$ is the set intersection $\cap$ with the neutral element $\mathbf{0} = \mathcal{I}$;
- The semantic function $[\![\cdot]\!]_{\mathsf{s}}$ is defined by $[\![\boldsymbol{T}]\!]_{\mathsf{s}} = \{ [\![\diamond \boldsymbol{T}]\!] \}$.

The splitting set, top, bottom and reduct of a first-order theory are defined analogically to the same notions for ontologies defined in the previous section. These are then naturally generalised to deal with sequences of first-order theories. For instance, the bottom of a sequence of first-order theories is the sequence of bottoms of theories in the sequences.

Definition 4.15 (Splitting Set, Bottom, Top and Reduct for First-Order Theories)**.** Let $\mathcal{T}$ be a first-order theory, $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$ a finite sequence of first-order theories and $U \subseteq \mathcal{P}$ a set of predicate symbols. We say that U is a

- *splitting set for $\mathcal{T}$* if for every formula $\phi \in \mathcal{T}$, if $\mathsf{pr}(\phi) \cap U \neq \emptyset$, then $\mathsf{pr}(\phi) \subseteq U$;
- *splitting set for $\boldsymbol{T}$* if for every $i < n$, U is a splitting set for $\mathcal{T}_i$.

$$* * * \; b_{U_0}(\mathcal{K}) \; * * *$$

$\mathsf{Commodity} \equiv (\exists \mathsf{HTSCode}.\top)$

$\mathsf{CherryTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020020'}\})$

$\mathsf{GrapeTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020010'}\})$

$\mathsf{CherryTomato} \sqsubseteq \mathsf{Tomato}$

$\mathsf{CherryTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$0\})$

$\mathsf{GrapeTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$40\})$

$\mathsf{CherryTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$50\})$

$\mathsf{GrapeTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$100\})$

$\mathsf{EdibleVegetable} \equiv (\exists \mathsf{HTSChapter}.\{\text{'07'}\})$

$\mathsf{Tomato} \equiv (\exists \mathsf{HTSHeading}.\{\text{'0702'}\})$

$\mathsf{Tomato} \sqsubseteq \mathsf{EdibleVegetable}$

$\mathsf{GrapeTomato} \sqsubseteq \mathsf{Tomato}$

$\mathsf{CherryTomato} \sqcap \mathsf{GrapeTomato} \sqsubseteq \bot$

$\mathsf{Bulk} \sqcap \mathsf{Prepackaged} \sqsubseteq \bot$

$\mathsf{ShpmtCommod}(s_1, c_1)$

$\mathsf{ShpmtImporter}(s_1, i_1)$

$\mathsf{ShpmtCommod}(s_2, c_2)$

$\mathsf{ShpmtImporter}(s_2, i_2)$

$\mathsf{ShpmtCountry}(s_2, portugal)$

$\mathsf{ShpmtCommod}(s_3, c_3)$

$\mathsf{ShpmtImporter}(s_3, i_3)$

$\mathsf{ShpmtCountry}(s_3, portugal)$

$\mathsf{ShpmtDeclHTSCode}(s_1, \text{'07020020'})$

$\mathsf{CherryTomato}(c_1) \quad \mathsf{Bulk}(c_1)$

$\mathsf{ShpmtDeclHTSCode}(s_2, \text{'07020020'})$

$\mathsf{CherryTomato}(c_2) \quad \mathsf{Prepackaged}(c_2)$

$\mathsf{ShpmtDeclHTSCode}(s_3, \text{'07020010'})$

$\mathsf{GrapeTomato}(c_3) \quad \mathsf{Bulk}(c_3)$

$\mathsf{ShpmtProducer}(s_3, p_1)$

$$* * * \; t_{U_0}(b_{U_1}(\mathcal{K})) \; * * *$$

$\mathsf{AdmissibleImporter}(\mathbf{x}) \leftarrow \sim\mathsf{SuspectedBadGuy}(\mathbf{x}).$

$\mathsf{SuspectedBadGuy}(i_1).$

$\mathsf{ApprovedImporterOf}(i_2, \mathbf{x}) \leftarrow \mathsf{EdibleVegetable}(\mathbf{x}).$

$\mathsf{ApprovedImporterOf}(i_3, \mathbf{x}) \leftarrow \mathsf{GrapeTomato}(\mathbf{x}).$

$$* * * \; t_{U_1}(b_{U_2}(\mathcal{K})) \; * * *$$

$\mathsf{EURegisteredProducer} \equiv (\exists \mathsf{RegisteredProducer}.\mathsf{EUCountry})$

$\mathsf{LowRiskEUCommodity} \equiv (\exists \mathsf{ExpeditableImporter}.\top) \sqcap (\exists \mathsf{CommodCountry}.\mathsf{EUCountry})$

$\mathsf{CommodCountry}(\mathbf{x}, \mathbf{y}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{z}, \mathbf{x}), \mathsf{ShpmtCountry}(\mathbf{z}, \mathbf{y}).$

$\mathsf{ExpeditableImporter}(\mathbf{x}, \mathbf{y}) \leftarrow \mathsf{AdmissibleImporter}(\mathbf{y}), \mathsf{ApprovedImporterOf}(\mathbf{y}, \mathbf{x}).$

$\mathsf{RegisteredProducer}(p_1, portugal)$

$\mathsf{RegisteredProducer}(p_2, slovakia)$

$\mathsf{EUCountry}(portugal)$

$\mathsf{EUCountry}(slovakia)$

$$* * * \; t_{U_2}(b_{U_3}(\mathcal{K})) \; * * *$$

$\mathsf{CompliantShpmt}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{HTSCode}(\mathbf{y}, \mathbf{z}), \mathsf{ShpmtDeclHTSCode}(\mathbf{x}, \mathbf{z}).$

$\mathsf{RandomInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{Random}(\mathbf{y}).$

$\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{RandomInspection}(\mathbf{x}).$

$\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \sim\mathsf{LowRiskEUCommodity}(\mathbf{y}).$

$\mathsf{FullInspection}(\mathbf{x}) \leftarrow \sim\mathsf{CompliantShpmt}(\mathbf{x}).$

$\mathsf{FullInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{Tomato}(\mathbf{y}), \mathsf{ShpmtCountry}(\mathbf{x}, slovakia).$

Figure 4.2: Layers of the MKNF knowledge base for Cargo Imports

The *bottom* and *top of $\mathcal{T}$ relative to U* are the theories

$$b_U(\mathcal{T}) = \{ \phi \in \mathcal{T} \mid \mathsf{pr}(\phi) \subseteq U \} \qquad \text{and} \qquad t_U(\mathcal{T}) = \mathcal{O} \setminus b_U(\mathcal{T}) \enspace .$$

The *bottom* and *top of $\boldsymbol{T}$ relative to U* are the sequences of theories

$$b_U(\boldsymbol{T}) = \langle b_U(\mathcal{T}_i) \rangle_{i<n} \qquad \text{and} \qquad t_U(\boldsymbol{T}) = \langle t_U(\mathcal{T}_i) \rangle_{i<n} \enspace .$$

Given some $\mathcal{X} \in \mathcal{M}$, the *reduct of $\mathcal{T}$ relative to U and $\mathcal{X}$* is $e_U(\mathcal{T}, \mathcal{X}) = t_U(\mathcal{T})$ and the *reduct of $\boldsymbol{T}$ relative to U and $\mathcal{X}$* is $e_U(\boldsymbol{T}, \mathcal{X}) = t_U(\boldsymbol{T})$.

Now that these definitions are established, we can use the generic definitions of splitting properties from Section 4.1 and instantiate them for first-order updates. Since every sequence of first-order theories has a single set of models, the resulting properties are simpler than their general form. Particularly, given a sequence of first-order theories $\boldsymbol{T}$ and a splitting set U for $\boldsymbol{T}$, the splitting set property requires that

$$[\![\diamond \boldsymbol{T}]\!] = [\![\diamond b_U(\boldsymbol{T})]\!] \cap [\![\diamond t_U(\boldsymbol{T})]\!] \enspace .$$

Similarly, given a splitting sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ for $\boldsymbol{T}$, the splitting sequence property requires that

$$[\![\diamond \boldsymbol{T}]\!] = [\![\diamond b_{U_0}(\boldsymbol{T})]\!] \cap \bigcap_{\alpha+1<\mu} [\![\diamond t_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{T}))]\!] \enspace .$$

If we pick Winslett's first-order operator as $\diamond$, then both of these properties are satisfied.

Theorem 4.16 (Splitting Theorem for Winslett's First-Order Update Operator)**.** *The semantics for sequences of first-order theories induced by Winslett's first-order operator $\diamond_{\mathsf{W}}$ satisfies the splitting set and splitting sequence properties.*

Proof. See Appendix C, page 211. □

However, as illustrated in the following example, many first-order update operators characterised by order assignments do not satisfy the splitting properties.

Example 4.17 (Violation of Splitting Properties by First-Order Update Operators)**.** Consider the sequence of theories $\boldsymbol{T} = \langle \mathcal{T}_0, \mathcal{T}_1 \rangle$ where $\mathcal{T}_0 = \{ p, q \}$ and $\mathcal{T}_1 = \{ \neg p, \neg q \}$ for some ABox assertions p and q. Furthermore, suppose that U is a set of predicate symbols such that $\mathsf{pr}(p) \subseteq U$ and $\mathsf{pr}(q) \subseteq \mathcal{P} \setminus U$. It follows that U is a splitting sequence for $\boldsymbol{T}$, $b_U(\boldsymbol{T}) = \langle \{ p \}, \{ \neg p \} \rangle$ and $t_U(\boldsymbol{T}) = \langle \{ q \}, \{ \neg q \} \rangle$. Now take a first-order update operator $\diamond$ characterised by a faithful order assignment ω such that for some ABox assertion r the following holds:

- for all interpretations I, J, K with $I \models p \wedge q$, $J \models \neg p \wedge \neg q$, $K \models \neg p \wedge \neg q$ the following holds:

$$J <^I_\omega K \qquad \text{if and only if} \qquad J \models r \wedge K \not\models r \enspace .$$

- for all interpretations I, J, K with $I \models p \wedge \neg q$, $J \models \neg p \wedge \neg q$, $K \models \neg p \wedge \neg q$ the following holds:

$$J <^I_\omega K \qquad \text{if and only if} \qquad J = I \wedge K \neq I \enspace .$$

- for all interpretations I, J, K with $I \models \neg p \wedge q$, $J \models \neg p \wedge \neg q$, $K \models \neg p \wedge \neg q$ the following holds:

$$J <^I_\omega K \qquad \text{if and only if} \qquad J = I \wedge K \neq I \ .$$

It follows from these relationships that

$$[\![\diamond \boldsymbol{T}]\!] = [\![\{ p, q \} \diamond \{ \neg p, \neg q \}]\!] \models r$$

while

$$[\![\diamond b_U(\boldsymbol{T})]\!] \cap [\![\diamond t_U(\boldsymbol{T})]\!] = [\![\{ p \} \diamond \{ \neg p \}]\!] \cap [\![\{ q \} \diamond \{ \neg q \}]\!] \not\models r \ .$$

These two observations are in conflict with the splitting set property.

Note that a first-order update operator that violates the splitting set property must be domain-dependent, i.e. the preorder assignment it is based on encodes some domain knowledge by causing a change in the interpretation of one predicate symbol to affect other predicate symbols. For instance, the assignment ω in the above example prefers interpretations that entail r over those that don't when making a transition from a state when both p and q were true to a state where they are both false, but not under other circumstances. In this way, ω encodes some domain-specific relationships between p, q and r. General-purpose operators, such as Winslett's, that treat all predicate symbols uniformly, do adhere to the splitting properties, so updates in syntactically independent parts of an ontology are also semantically independent from one another.

4.2.3 Rule Updates

Given a rule update semantics S, the splitting properties for DLPs can be derived from the generic ones as follows:

- The set of theories $\mathcal{T}$ contains all DLPs;
- The set of semantic structures $\mathcal{S}$ coincides with the set of all ASP interpretations;
- The operator $\uplus$ is the set union $\cup$ and its neutral element is $\mathbf{0} = \emptyset$;
- The semantic function $[\![\cdot]\!]_{\mathsf{S}}$ is as defined by the rule update semantics S, i.e. it returns the set of S-stable models of the argument DLP.

Since rule update semantics work with ASP interpretations instead of MKNF interpretations, we define program reduct w.r.t. an ASP interpretation as the program reduct w.r.t. the corresponding MKNF interpretation (c.f. Definition 2.18). Note that despite the different definition, the resulting concept is in line with the same notion in (Lifschitz and Turner, 1994).

Definition 4.18 (Reduct of a Program w.r.t. a Splitting Set)**.** Let $\mathcal{P}$ be a program, $U \subseteq \mathcal{P}$ a set of predicate symbols, J an ASP interpretation and $\mathcal{M}$ the MKNF interpretation corresponding to J. We define the *reduct of $\mathcal{P}$ w.r.t. U and J* as $e_U(\mathcal{P}, J) = e_U(\mathcal{P}, \mathcal{M})$.

The splitting set, bottom, top and reduct of a DLP are now straightforward adaptations of the same notions for single programs.

Definition 4.19 (Splitting Set, Bottom, Top and Reduct for a DLP)**.** Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP and U a set of predicate symbols. We say that *U is a splitting set for $\boldsymbol{P}$* if for all $i < n$, U is a splitting set for $\mathcal{P}_i$.

The *bottom* and *top of $\boldsymbol{P}$ relative to U* are defined as $b_U(\boldsymbol{P}) = \langle b_U(\mathcal{P}_i)\rangle_{i<n}$ and $t_U(\boldsymbol{P}) = \langle t_U(\mathcal{P}_i)\rangle_{i<n}$, respectively. Given an ASP interpretation J, the *reduct of $\boldsymbol{P}$ relative to U and J* is defined as $e_U(\boldsymbol{P}, J) = \langle e_U(\mathcal{P}_i, J)\rangle_{i<n}$.

Given these definitions, the splitting properties for a rule update semantics S are directly derived from the generic ones defined in Section 4.1. Intuitively, given a DLP $\boldsymbol{P}$ and a splitting set U, the splitting set property requires that every $J \in [\![\boldsymbol{P}]\!]_{\mathsf{S}}$ be the union of some $J' \in [\![b_U(\boldsymbol{P})]\!]_{\mathsf{S}}$ and some $J'' = [\![e_U(\boldsymbol{P}, J')]\!]_{\mathsf{S}}$. The splitting sequence property generalises the same requirement to splitting sequences. Many rule update semantics, such as the causal rejection-based rule update semantics, can be shown to satisfy both splitting properties.

Theorem 4.20 (Splitting Theorem for Rule Update Semantics). *The rule update semantics AS, JU, DS and RD satisfy the splitting set and splitting sequence properties.*

Proof (sketch). See Appendix C, page 212. □

Note, however, that not all rule update semantics satisfy the splitting properties. This is demonstrated for the RVS-semantics in the following example:

Example 4.21 (Violation of Splitting Properties by RVS-Semantics). Consider the DLP

$$\boldsymbol{P} = \langle \{\, p., q \leftarrow p. \,\}, \{\, \neg q \leftarrow p. \,\}\rangle \ .$$

Suppose that U is a set of predicate symbols such that $\mathsf{pr}(p) \subseteq U$ and $\mathsf{pr}(q) \subseteq \mathcal{P} \setminus U$. It follows that U is a splitting sequence of $\boldsymbol{P}$ and we have $b_U(\boldsymbol{P}) = \langle \{\, p. \,\}, \emptyset\rangle$ and $t_U(\boldsymbol{P}) = \langle \{\, q \leftarrow p. \,\}, \{\, \neg q \leftarrow p. \,\}\rangle$.

The DLP $\boldsymbol{P}$ has two RVS-models: $\{\, p, \neg q \,\}$ and $\emptyset$. However, the only RVS-model of $b_U(\boldsymbol{P})$ is $\{\, p \,\}$ and $e_U(\boldsymbol{P}, \{\, p \,\}) = \langle \{\, q. \,\}, \{\, \neg q. \,\}\rangle$ has the unique RVS-model $\{\, \neg q \,\}$. The splitting set property is thus violated because the RVS-model $\emptyset$ of $\boldsymbol{P}$ is not the union of an RVS-model of the bottom of $\boldsymbol{P}$ and of an RVS-model of the corresponding reduct of $\boldsymbol{P}$.

4.3 Splitting-Based Updates of MKNF Knowledge Bases

We are now ready to utilise the splitting properties as a foundation for a hybrid update semantics. In the following we identify a class of DMKBs in which the interaction between ontology axioms and rules is limited and show how splitting enables us to use a given first-order update operator and a given rule update semantics to define updates of such DMKBs.

Throughout the remainder of this chapter we assume that some first-order update operator $\diamond$ and some rule update semantics S are given and fixed.

We start by defining a *basic DMKB* which can be handled by $\diamond$ or S alone. More specifically, we allow a basic DMKB to contain

a) arbitrary ontological axioms but no rules except for positive facts (i.e. rules with an empty body and a single objective literal in the head);

b) arbitrary rules but no ontological axioms whatsoever.

Formally:

Definition 4.22 (Basic DMKB). We say that a hybrid knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ is *ontology-based* if $\mathcal{P}$ is a consistent set of positive facts; *rule-based* if $\mathcal{O}$ is empty; *basic* if it is either ontology- or rule-based.

A DMKB $\boldsymbol{K} = \langle \mathcal{K}_i \rangle_{i<n}$ is *ontology-based* if for all $i < n$, $\mathcal{K}_i$ is ontology-based; *rule-based* if for all $i < n$, $\mathcal{K}_i$ is rule-based; *basic* if it is either ontology- or rule-based.

Ontology-based DMKBs can be handled by the first-order update operator $\diamond$ while rule-based DMKBs can be updated using the rule update semantics S. This can be formalised as follows:

Definition 4.23 (Update Semantics for Basic DMKBs)**.** Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$ be a basic DMKB. An MKNF interpretation $\mathcal{M}$ is a $(\diamond, \mathsf{S})$*-dynamic MKNF model* of $\boldsymbol{K}$ if either

a) $\boldsymbol{K}$ is ontology-based and $\mathcal{M} = [\![\diamond\langle \kappa(\mathcal{O}_i) \cup \{ l \mid (l.) \in \mathcal{P}_i \} \rangle_{i<n}]\!]$, or
b) $\boldsymbol{K}$ is rule-based and $\mathcal{M}$ corresponds to some $J \in [\![\langle \mathcal{P}_i \rangle_{i<n}]\!]_{\mathsf{S}}$.

By allowing programs in an ontology-based DMKB to contain positive facts, we pave the way towards extending the class of basic DMKBs to a much larger class for which we define an update semantics through splitting. Turning back to the splitting-based update semantics for DMKBs, the central idea is that if a DMKB $\boldsymbol{K}$ can be split into multiple layers, each of which consists of a basic DMKB, then the above defined update semantics for basic DMKBs can be used to assign a semantics to $\boldsymbol{K}$. We thus define the splitting set, bottom, top and reduct for DMKBs as follows:

Definition 4.24 (Splitting Set, Bottom, Top and Reduct for a DMKB)**.** Let $\boldsymbol{K} = \langle \mathcal{K}_i \rangle_{i<n}$ be a DMKB and U a set of predicate symbols. We say that U *is a splitting set for* $\boldsymbol{K}$ if for all $i < n$, U is a splitting set for $\mathcal{K}_i$.

The *bottom* and *top of* $\boldsymbol{K}$ *relative to* U are defined as $b_U(\boldsymbol{K}) = \langle b_U(\mathcal{K}_i) \rangle_{i<n}$ and $t_U(\boldsymbol{K}) = \langle t_U(\mathcal{K}_i) \rangle_{i<n}$, respectively.

Given some $\mathcal{X} \in \mathfrak{M}$, the *reduct of* $\boldsymbol{K}$ *relative to* U *and* $\mathcal{X}$ is defined as $e_U(\boldsymbol{K}, \mathcal{X}) = \langle e_U(\mathcal{K}_i, \mathcal{X}) \rangle_{i<n}$.

With these definitions in place, we can instantiate the generic definitions from Section 4.1 and obtain the definition of a splitting sequence as well as of a solution w.r.t. a splitting set and splitting sequence. But we still need to make sure that after splitting, the obtained DMKBs are basic. In case of a single splitting set U this amounts to requiring that the bottom layer $b_U(\boldsymbol{K})$ be a basic DMKB and the reduct $e_U(\boldsymbol{K}, \mathcal{X})$ also be a basic DMKB. Similarly, for a splitting sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ we need to make sure that $b_{U_0}(\boldsymbol{K})$ is a basic DMKB and for every ordinal α such that $\alpha + 1 < \mu$, $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{X})$ is a basic DMKB. The following definition establishes the notion of a *layering splitting sequence* by requiring exactly these conditions given an arbitrary choice of $\mathcal{X}$.

Definition 4.25 (Layering Splitting Sequence)**.** Let $\boldsymbol{K}$ be a DMKB and $U = \langle U_\alpha \rangle_{\alpha<\mu}$ a splitting sequence for $\boldsymbol{K}$. We say that U is a *layering splitting sequence for* $\boldsymbol{K}$ if $b_{U_0}(\boldsymbol{K})$ is a basic DMKB and for any ordinal α such that $\alpha+1 < \mu$ and every $\mathcal{X} \in \mathfrak{M}$, $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{X})$ is also a basic DMKB. We say that $\boldsymbol{K}$ is *layered* if some layering splitting sequence for $\boldsymbol{K}$ exists.

The definition of a solution to a DMKB $\boldsymbol{K}$ w.r.t. a layering splitting sequence is an instantiation of the abstract definition in Section 4.1. We formulate it here for the sake of completeness and use solutions to establish the concept of a $(\diamond, \mathsf{S})$*-dynamic MKNF model of* $\boldsymbol{K}$ *w.r.t. a layering splitting sequence*.

Definition 4.26 (Solution to a Layered DMKB)**.** Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$ be a DMKB and $\boldsymbol{U}$ a layering splitting sequence for $\boldsymbol{K}$. A *solution to* $\boldsymbol{K}$ *w.r.t.* $\boldsymbol{U}$ is a sequence of MKNF interpretations $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ such that

1. $\mathcal{X}_0$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $b_{U_0}(\boldsymbol{K})$;
2. For any ordinal α such that $\alpha + 1 < \mu$, $\mathcal{X}_{\alpha+1}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right) \quad ;$$

3. For any limit ordinal α, $\mathcal{X}_\alpha = \mathcal{I}$;
4. $\bigcap_{\alpha<\mu} \mathcal{X}_\alpha \neq \emptyset$.

We say that $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-*dynamic MKNF model of* $\boldsymbol{K}$ *w.r.t.* $\boldsymbol{U}$ if $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ for some solution $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$.

However, the $(\diamond, \mathsf{S})$-dynamic MKNF models defined above depend on a particular splitting sequence and there is no guarantee that under a different splitting, the same models will be obtained. In the following we introduce conditions under which these models are independent of a splitting sequence. In particular, we need to assume the following properties of $\diamond$ and S:

1. Splitting properties: Both $\diamond$ and S must satisfy the splitting properties. If this were not the case, then solutions might depend on a splitting sequence even for basic DMKBs.

2. Language conservation: Language conservation (c.f. Definitions 2.42 and 2.69) must also be satisfied by both $\diamond$ and S, for otherwise they may interfere with one another when used to update syntactically unrelated layers of the DMKB.

3. Fact update: Finally, since DMKBs consisting of a sequence of consistent sets of facts are classified by Definition 4.22 as both ontology- and rule-based, their semantics is given by both $\diamond$ and by S. This ambivalence is unavoidable if we want to allow both an ontology and a rule layer to contain only facts, and in particular to be simply empty. Nevertheless, if the semantics assigned to such sequences of fact bases by $\diamond$ differs from the semantics assigned by S, the resulting hybrid update semantics cannot generalise $\diamond$ nor S. In order to avoid such anomalies, we assume that both $\diamond$ and S respect fact update (c.f. Definitions 2.44 and 2.71).

Note that if we consider Winslett's operator for performing ontology updates and some causal rejection-based rule update semantics for performing rule updates, all of these properties are satisfied (c.f. Theorems 4.16, 4.20, 2.43, 2.70, 2.45 and 2.72). Under these assumptions, we can show that $(\diamond, \mathsf{S})$-dynamic MKNF models are independent of the choice of the layering splitting sequence.

Proposition 4.27 (Independence of Splitting Sequence)**.** *Let* $\boldsymbol{U}$, $\boldsymbol{V}$ *be layering splitting sequences for a DMKB* $\boldsymbol{K}$*. If both* $\diamond$ *and* S *have the splitting sequence property, conserve the language and respect fact update, then* $\mathcal{M}$ *is a* $(\diamond, \mathsf{S})$*-dynamic MKNF model of* $\boldsymbol{K}$ *w.r.t.* $\boldsymbol{U}$ *if and only if* $\mathcal{M}$ *is a* $(\diamond, \mathsf{S})$*-dynamic MKNF model of* $\boldsymbol{K}$ *w.r.t.* $\boldsymbol{V}$*.*

Proof. See Appendix C, page 217. □

We can now safely introduce the dynamic MKNF model of a layered DMKB, independent of a particular layering splitting sequence:

Definition 4.28 (Update Semantics for Layered DMKBs)**.** Suppose that both $\diamond$ and S have the splitting sequence property, conserve the language and respect fact update, and let $\boldsymbol{K}$ be a layered DMKB. An MKNF interpretation $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-*dynamic MKNF model of* $\boldsymbol{K}$ if $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ w.r.t. some layering splitting sequence.

Note that since $\langle \mathcal{P} \rangle$ is a layering splitting sequence for any basic DMKB, it follows from Proposition 4.27 that the above definition is compatible with the previously defined semantics for basic DMKBs (c.f. Definition 4.23).

4.4 Properties and Use

The purpose of this section is twofold. First, we establish the basic properties of the defined hybrid update semantics, showing that it is faithful to the (static) semantics of MKNF knowledge bases as well as the first-order update operator and rule update semantics it is based upon. We also prove that it respects one of the most widely accepted principles underlying update semantics in general, the principle of primacy of new information, and is in line with the hybrid update semantics introduced in Chapter 3. Our second goal is to illustrate its usefulness by considering updates of the MKNF knowledge base about Cargo Imports presented in Example 4.1.

We assume throughout this section that both the first-order update operator $\diamond$ and the rule update semantics S have the splitting sequence property, conserve the language and respect fact update.

We first define two basic properties of a rule update semantics that we need to assume in some of the theoretical results. The first property requires faithfulness of a rule update semantics to stable models while the second is concerned with respect for primacy of new information. Both properties are satisfied by most existing rule update semantics.

Definition 4.29 (Properties of Rule Update Semantics)**.** Let S be a rule update semantics. We say that S

- *is faithful to the stable models semantics* if for every program $\mathcal{P}$, an ASP interpretation J is a stable model of $\mathcal{P}$ if and only if J is an S-model of the DLP $\langle \mathcal{P} \rangle$;
- *respects primacy of new information* if for every DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ such that $n > 0$ and every S-model J of $\boldsymbol{P}$ it holds that $J \models \mathcal{P}_{n-1}$.

Our first formal result about the hybrid update semantics shows that it is faithful to the semantics of MKNF knowledge bases. For this to work, we need to assume that the rule update semantics is faithful to the stable models semantics.

Theorem 4.30 (Faithfulness w.r.t. MKNF Knowledge Bases)**.** *Suppose that S is faithful to the stable models semantics and let $\langle \mathcal{K} \rangle$ be a layered DMKB. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\langle \mathcal{K} \rangle$.*

Proof. See Appendix C, page 220. □

An immediate consequence of this result and of Propositions 2.17 and 2.19 is that the defined update semantics is also faithful w.r.t. ontologies and stable models.

Furthermore, the semantics is faithful to the first-order update operator $\diamond$ and the rule update semantics S that it is based on.

Theorem 4.31 (Faithfulness w.r.t. First-Order Update Operator)**.** *Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \emptyset) \rangle_{i<n}$ be a DMKB. An MKNF interpretation $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M} = [\![\Diamond \langle \mathcal{O}_i \rangle_{i<n}]\!]$.*

Proof. See Appendix C, page 220. □

Theorem 4.32 (Faithfulness w.r.t. Rule Update Semantics)**.** *Let* $\boldsymbol{K} = \langle(\emptyset, \mathcal{P}_i)\rangle_{i<n}$ *be a DMKB. If* J *is an* S*-model of* $\langle\mathcal{P}_i\rangle_{i<n}$*, then the MKNF interpretation corresponding to* J *is a* $(\diamond, \mathsf{S})$*-dynamic MKNF model of* $\boldsymbol{K}$*. If* $\mathcal{M}$ *is a* $(\diamond, \mathsf{S})$*-dynamic MKNF model of* $\boldsymbol{K}$*, then the ASP interpretation corresponding to* $\mathcal{M}$ *is an* S*-model of* $\langle\mathcal{P}_i\rangle_{i<n}$*.*

Proof. See Appendix C, page 220. □

Similarly as the update semantics in Chapter 3, assuming that $\diamond$ satisfies (FO1) and S respects primacy of new information, our semantics also respects it.

Theorem 4.33 (Primacy of New Information)**.** *Suppose that* $\diamond$ *satisfies (FO1) and* S *respects primacy of new information and let* $\boldsymbol{K} = \langle\mathcal{K}_i\rangle_{i<n}$ *be a layered DMKB such that* $n > 0$*. If* $\mathcal{M}$ *is a* $(\diamond, \mathsf{S})$*-dynamic MKNF model of* $\boldsymbol{K}$*, then* $\mathcal{M} \models \kappa(\mathcal{K}_{n-1})$*.*

Proof. See Appendix C, page 221. □

Besides, the update semantics is compatible with the semantics from Chapter 3 – both of them provide the same results when applied to *layered* DMKBs *with static rules*, i.e. to the class of DMKBs that they can both handle.

Theorem 4.34 (Compatibility with Update Semantics from Chapter 3)**.** *Suppose that* $\diamond$ *satisfies (FO2.*$\top$*) and (FO8.2) and that* S *is faithful to the stable models semantics. Let* $\boldsymbol{K}$ *be a layered DMKB with static rules. An MKNF interpretation* $\mathcal{M}$ *is a* $\diamond$*-dynamic MKNF model of* $\boldsymbol{K}$ *if and only if* $\mathcal{M}$ *is a* $(\diamond, \mathsf{S})$*-dynamic MKNF model of* $\boldsymbol{K}$*.*

Proof. See Appendix C, page 227. □

The following example illustrates how the semantics can be used in the Cargo Imports domain to incorporate new, conflicting information into an MKNF knowledge base.

Example 4.35 (Updating the Cargo Imports Knowledge Base)**.** The MKNF knowledge base $\mathcal{K}$ in Figure 4.1 has a single MKNF model $\mathcal{M}$. We shortly summarise what is entailed by this model. First, since the shipments s_1, s_2, s_3 differ in the kind of tomatoes or in their packaging, each of them is assigned a different tariff charge. HTS codes of commodities inside all three shipments match the declared HTS codes, so CompliantShipment(s_i) is entailed for all i. The rules for importers imply that while both AdmissibleImporter(i_2) and AdmissibleImporter(i_3) are true, AdmissibleImporter(i_1) is not true because i_1 is a suspected bad guy. It also follows that ApprovedImporterOf(i_2, c_2) and ApprovedImporterOf(i_3, c_3) hold and, because of that, ExpeditableImporter(c_2, i_2) and ExpeditableImporter(c_3, i_3) are also true. Both shipments s_2 and s_3 come from a European country, so c_2 and c_3 belong to LowRiskEUCommodity. But this is not true for c_1 since there is no expeditable importer for it. Consequently, PartialInspection(s_1) holds.

We now consider an update caused by several independent events in order to illustrate different aspects of the hybrid update semantics. We assume to be using Winslett's first-order operator $\diamond_{\mathsf{W}}$ to deal with ontology updates and the RD-semantics to deal with rule updates.

Suppose that during the partial inspection of s_1, grape tomatoes are found instead of cherry tomatoes. Second, we suppose that i_2 is no longer an approved importer for any kind of tomatoes due to a history of mis-filing. Third, due to a rat infestation on the boat with shipment s_3, c_3 is no longer considered a low risk commodity. Finally, because of workload constraints, partial inspections for shipments with commodities from a producer registered in a country of the European Union will be waived. These events lead to the following update $\mathcal{K}' = (\mathcal{O}', \mathcal{P}')$ where $\mathcal{O}'$ contains the assertions

$$\mathsf{GrapeTomato}(c_1) \ , \\ \neg\mathsf{LowRiskEUCommodity}(c_3)$$

as well as all TBox axioms from $\mathcal{O}$,[2] and $\mathcal{P}'$ contains the following rules:[3]

$$\sim\mathsf{ApprovedImporterOf}(i_2, \mathbf{x}) \leftarrow \mathsf{Tomato}(\mathbf{x}).$$
$$\sim\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtProducer}(\mathbf{x}, \mathbf{y}), \mathsf{EURegisteredProducer}(\mathbf{y}).$$

Note that the splitting sequence $\boldsymbol{U}$ defined in Example 4.14 is a layering splitting sequence for the DMKB $\langle \mathcal{K}, \mathcal{K}' \rangle$. The four layers of $\mathcal{K}$ and $\mathcal{K}'$ are listed in Figures 4.2 and 4.3, respectively. The first layer $\langle b_{U_0}(\mathcal{K}), b_{U_0}(\mathcal{K}') \rangle$ contains only ontology axioms and so is ontology-based. The second and fourth layers $\langle t_{U_0}(b_{U_1}(\mathcal{K})), t_{U_0}(b_{U_1}(\mathcal{K}')) \rangle$ and $\langle t_{U_2}(b_{U_3}(\mathcal{K})), t_{U_2}(b_{U_3}(\mathcal{K}')) \rangle$ contain only rules and so are rule-based. On the other hand, the third layer $\langle t_{U_1}(b_{U_2}(\mathcal{K})), t_{U_1}(b_{U_2}(\mathcal{K}')) \rangle$ contains a mixture of rules and ontology axioms but all the rules have positive heads and predicate symbols of body literals belong to U_1, so the reduct of the third layer will necessarily be ontology-based.

In order to arrive at a $(\diamond_{\mathsf{W}}, \mathsf{RD})$-dynamic MKNF model of $\langle \mathcal{K}, \mathcal{K}' \rangle$ with respect to $\boldsymbol{U}$, a $(\diamond_{\mathsf{W}}, \mathsf{RD})$-dynamic MKNF model of each layer is determined separately and models of previous layers serve to "feed" information into the current layer.

In our case, we first find the model $\mathcal{X}_0$ of the first layer of $\mathcal{K}$ updated by the first layer of $\mathcal{K}'$. Due to the TBox axioms, this results in c_1 no longer being a member of CherryTomato. The HTS code of c_1 also changes to '07020010'. Note that the conflict between old and new knowledge is properly resolved by Winslett's operator.

Subsequently, the RD-semantics is used to find the model of the second layer of $\mathcal{K}$ updated by the second layer of $\mathcal{K}'$. This rule update results in i_2 no longer being an approved importer for c_2. As before, the conflict that arose is resolved by the RD-semantics.

Given the $(\diamond_{\mathsf{W}}, \mathsf{RD})$-dynamic MKNF models of the first two layers, the model of the third layer of $\mathcal{K}$ is now different because i_2 is no longer an expeditable importer of c_2. As a consequence, c_2 is no longer a member of the concept LowRiskEUCommodity. Also, due to the update in the third layer, c_3 is not a member of LowRiskEUCommodity. The conflicting situation is again resolved by Winslett's operator and results in the $(\diamond_{\mathsf{W}}, \mathsf{RD})$-dynamic MKNF model $\mathcal{X}_2$ of the third layer of $\langle \mathcal{K}, \mathcal{K}' \rangle$.

Finally, due to the changes in all three previous layers, the rules in the fourth layer imply that $\mathsf{CompliantShpmt}(s_1)$ does not hold and, as a consequence, $\mathsf{FullInspection}(s_1)$ holds. Also, $\mathsf{PartialInspection}(s_2)$ holds because c_2 is not a low risk commodity. But even though c_3 is also not a low risk commodity, $\mathsf{PartialInspection}(s_3)$ does not hold. This is due to the rule update of the fourth layer according to which the inspection of s_3 must be waived because s_3 comes from an EU registered producer. The resulting model $\mathcal{X}_3$ of the last layer is determined by the RD-semantics.

Note that the sequence $\langle \mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3 \rangle$ is a solution to $\langle \mathcal{K}, \mathcal{K}' \rangle$ w.r.t. the splitting sequence $\langle U_0, U_1, U_2, U_3 \rangle$ and the unique $(\diamond_{\mathsf{W}}, \mathsf{RD})$-dynamic MKNF model assigned to $\langle \mathcal{K}, \mathcal{K}' \rangle$ is the MKNF interpretation $\mathcal{M} = \mathcal{X}_0 \cap \mathcal{X}_1 \cap \mathcal{X}_2 \cap \mathcal{X}_3$.

[2]Due to the difficulties that Winslett's operator has with updating TBoxes (c.f. Example 1.8), we reinclude all TBox axioms from $\mathcal{O}$ in $\mathcal{O}'$ in order to keep them static throughout the example.

[3]We assume that all rule variables are grounded prior to applying our theory.

$$* * * \; b_{U_0}(\mathcal{K}') \; * * *$$

$\mathsf{Commodity} \equiv (\exists \mathsf{HTSCode}.\top)$
$\mathsf{CherryTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020020'}\})$
$\mathsf{GrapeTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020010'}\})$
$\mathsf{CherryTomato} \sqsubseteq \mathsf{Tomato}$
$\mathsf{CherryTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$0\})$
$\mathsf{GrapeTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$40\})$
$\mathsf{CherryTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$50\})$
$\mathsf{GrapeTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$100\})$
$\mathsf{EdibleVegetable} \equiv (\exists \mathsf{HTSChapter}.\{\text{'07'}\})$
$\mathsf{Tomato} \equiv (\exists \mathsf{HTSHeading}.\{\text{'0702'}\})$
$\mathsf{Tomato} \sqsubseteq \mathsf{EdibleVegetable}$
$\mathsf{GrapeTomato} \sqsubseteq \mathsf{Tomato}$
$\mathsf{CherryTomato} \sqcap \mathsf{GrapeTomato} \sqsubseteq \bot$
$\mathsf{Bulk} \sqcap \mathsf{Prepackaged} \sqsubseteq \bot$

$\mathsf{GrapeTomato}(c_1)$

$$* * * \; t_{U_0}(b_{U_1}(\mathcal{K}')) \; * * *$$

$\sim\mathsf{ApprovedImporterOf}(i_2, \mathbf{x}) \leftarrow \mathsf{Tomato}(\mathbf{x}).$

$$* * * \; t_{U_1}(b_{U_2}(\mathcal{K}')) \; * * *$$

$\mathsf{EURegisteredProducer} \equiv (\exists \mathsf{RegisteredProducer}.\mathsf{EUCountry})$
$\mathsf{LowRiskEUCommodity} \equiv (\exists \mathsf{ExpeditableImporter}.\top) \sqcap (\exists \mathsf{CommodCountry}.\mathsf{EUCountry})$

$\neg\mathsf{LowRiskEUCommodity}(c_3)$

$$* * * \; t_{U_2}(b_{U_3}(\mathcal{K}')) \; * * *$$

$\sim\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtProducer}(\mathbf{x}, \mathbf{y}), \mathsf{EURegisteredProducer}(\mathbf{y}).$

Figure 4.3: Layers of the update to the MKNF knowledge base for Cargo Imports

4.5 Discussion

We have introduced a hybrid update semantics for DMKBs consisting of ontology and rule layers and demonstrated its usefulness in a realistic scenario. The semantics is capable of performing non-trivial updates, automatically resolving conflicts in the expected manner, and propagating new information across the knowledge base. Its practical usefulness is underlined by the fact that the full expressivity of MKNF knowledge bases does not seem to be necessary in a number of use cases of hybrid knowledge. The separation of a hybrid knowledge base into distinct ontology and rule layers seems to be a natural way of controlling, from the perspective of a knowledge engineer, how the different types of knowledge interact.

One of the main technical advantages of this approach is that one can fully reuse existing ontology and rule update semantics without the need to modify them internally in any way. In particular, our construction works when the given ontology and rule update semantics satisfy the *splitting properties* which we formulated formally in Section 4.1, *conserve the language* and *respect fact update*. We have also shown that the static MKNF semantics as well as Winslett's first-order operator, commonly used for performing ontology updates, and many rule update semantics, such as the causal rejection-based ones, satisfy the splitting properties. These update semantics can thus be used to parametrise the defined update semantics for layered MKNF knowledge bases.

Furthermore, we proved that the semantics enjoys a number of desirable theoretical properties, namely that it is faithful to the semantics for MKNF knowledge bases as well as to the constituent ontology and rule update semantics. Also, in case both the ontology and rule update semantics respect primacy of new information, the resulting hybrid update semantics also respects it. Another interesting theoretical result is that the framework is in line with the update semantics for *DMKBs with static rules* presented in Chapter 3. In other words, when both semantics are applicable to a particular DMKB, they produce the same dynamic MKNF models for it.

5

Difficulties with Updates of Hybrid Knowledge Bases

> Chaos is found in greatest abundance wherever order is being sought. It always defeats order, because it is better organized.
>
> Terry Pratchett
> *Interesting Times*

In the previous two chapters we addressed the problem of hybrid updates in two constrained scenarios, each of them applicable to different use cases of hybrid knowledge bases. We showed that the proposed solutions are fully compatible with each other, i.e. when both are applicable to a particular DMKB, the provided result is the same. The natural question that comes to mind is: Can they be naturally extrapolated to arrive at a universal hybrid update semantics?

In this chapter we discuss the overall difficulties with finding such a universal semantics and argue that in order to define appropriate hybrid update semantics, more needs to be understood regarding TBox updates and the interconnection between belief and rule updates. This motivates us to turn away from directly addressing hybrid updates and concentrate, in the remainder of this thesis, on bridging the gap between belief and rule updates which may bring new insights in both these areas and provide means for suitably combining them.

The first set of issues in trying to define semantics for DMKBs stems from the fact that even when we constrain ourselves to ontologies or rules alone, we find serious problems without established solutions. We could see throughout Section 2.8 that existing rule update semantics, though they share a common basis, differ significantly from one another, even on very simple examples. Currently there does not seem to exist any consensus within the community regarding which semantics is best under what circumstances. Rather, it seems that this area of research is still waiting for a breakthrough that would clarify how rule update semantics are related to one another as well as to belief change postulates and operators. In addition, work on updates of disjunctive programs is virtually non-existent.

On the ontology side, in Section 2.7 we briefly described the problems with expressibility of the updated ontology in the original DL. In case of ABox updates, it has been shown that in certain DLs expressibility of the updated ABox is guaranteed. Nevertheless, computational cost remains an important issue, especially in the case of expressive DLs where the ABox may grow exponentially with each update.

But the difficulties with ontology updates are most evident when we consider *TBox updates*. In Example 1.8 we have shown that Winslett's update operator, most commonly used to deal with ABox updates, leads to rather unexpected results when used to update TBoxes. Furthermore, intuition suggests that this behaviour is not limited to Winslett's operator. We devote Section 5.1 to formalising this intuition and showing that model-based update operators are incompatible with certain expectations regarding TBox updates. This suggests that methods other than those studied in the area of belief updates need to be explored in order to deal with TBox updates. Currently there is still very little research addressing this topic, an exception being (Calvanese et al., 2010).

Furthermore, the capital differences between methods underlying ontology and rule updates make their semantic combination very troublesome. We illustrate some of the related problems on examples in Section 5.2, pointing at how this hinders a definition of a general-purpose hybrid update semantics.

Given all these difficulties, in Section 5.3 we justify our decision to devote the following chapters to bridging the gap between belief and rule updates by looking for semantic counterparts of traditional syntax-based rule update semantics.

5.1 Problems with TBox Updates

We illustrated in Example 1.8 that Winslett's operator does not seem to be appropriate for performing TBox updates. This has motivated Calvanese et al. (2010) to abandon model-based belief updates altogether and use other types of operators for updating TBoxes, namely the formula-based update operators that bear a strong resemblance to belief revision operators (c.f. Section 2.5.3 for more details).

In the present section we generalise the case made by Example 1.8 to argue that not only Winslett's operator, but any model-based belief update operator necessarily leads to problematic behaviour when used for updating TBoxes. We do so by proving an impossibility result, showing that certain natural expectations from TBox update operators are incompatible with fundamental belief update postulates.

Let us first extract the essence of the problematic behaviour from Example 1.8. Suppose that A, B and C are concept names. Then the following holds:

$$\{ B \sqsubseteq A \} \diamond_{\mathsf{W}} \{ C \sqsubseteq B \} \not\models B \sqsubseteq A \enspace .$$

In other words, though we would not expect the update to influence the original knowledge, Winslett's operator $\diamond_{\mathsf{W}}$ actually weakens the original axiom.

In order to extend this result to other model-based belief update operators, we need to consider two slightly modified instances of the above situation. In both of them we perform the same update

$$\mathcal{U} = \{ C \sqsubseteq B_1, C \sqsubseteq B_2, \{ a \} \sqsubseteq C \} \ . \tag{5.1}$$

The purpose of $\mathcal{U}$ is to introduce C as a new subconcept of concepts B_1 and B_2 and to assert that a belongs to C. Now consider two independent initial situations

$$\mathcal{T}_1 = \{ B_1 \sqsubseteq A \} \qquad \text{and} \qquad \mathcal{T}_2 = \{ B_2 \sqsubseteq \neg A \} \ . \tag{5.2}$$

The expectation is that since $\mathcal{U}$ only introduces the new concept C, it should not affect the original information in $\mathcal{T}_1$ nor in $\mathcal{T}_2$. More formally, it should hold that

$$\mathcal{T}_1 \diamond \mathcal{U} \models \mathcal{T}_1 \qquad \text{and} \qquad \mathcal{T}_2 \diamond \mathcal{U} \models \mathcal{T}_2 \ . \tag{5.3}$$

As it turns out, this is impossible if the update operator $\diamond$ satisfies postulates (FO1), (FO3), and (FO8.2), propositional counterparts of which are generally accepted in the belief update community (Herzig and Rifi, 1999).

We first prove the following auxiliary statement:

Lemma 5.1. *Let $\diamond$ be a first-order update operator that satisfies postulate (FO8.2). Then the following holds for all first-order theories $\mathcal{T}, \mathcal{S}, \mathcal{U}, \mathcal{V}$:*

$$\textit{If } \mathcal{T} \diamond \mathcal{U} \models \mathcal{V}, \textit{ then } (\mathcal{T} \cup \mathcal{S}) \diamond \mathcal{U} \models \mathcal{V}.$$

Proof. The claim follows directly by applying (FO8.2): from $\mathcal{T} \cup \mathcal{S} \models \mathcal{T}$ we obtain $(\mathcal{T} \cup \mathcal{S}) \diamond \mathcal{U} \models \mathcal{T} \diamond \mathcal{U}$ and since by the assumption $\mathcal{T} \diamond \mathcal{U} \models \mathcal{V}$, it follows that $(\mathcal{T} \cup \mathcal{S}) \diamond \mathcal{U} \models \mathcal{V}$ by transitivity of entailment. □

The impossibility result is now easy to establish.

Theorem 5.2. *Let $\diamond$ be a first-order update operator that satisfies postulates (FO1), (FO3) and (FO8.2) and $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{U}$ be as in (5.2) and (5.1) for some concept names A, B_1, B_2, C. Then condition (5.3) is violated.*

Proof. We prove by contradiction. Suppose that condition (5.3) is satisfied. The following then follows from Lemma 5.1:[1]

$$(\kappa(\mathcal{T}_1) \cup \kappa(\mathcal{T}_2)) \diamond \kappa(\mathcal{U}) \models \kappa(\mathcal{T}_1) \qquad \text{and} \qquad (\kappa(\mathcal{T}_1) \cup \kappa(\mathcal{T}_2)) \diamond \kappa(\mathcal{U}) \models \kappa(\mathcal{T}_2) \ .$$

It also follows by (FO1) that

$$(\kappa(\mathcal{T}_1) \cup \kappa(\mathcal{T}_2)) \diamond \kappa(\mathcal{U}) \models \kappa(\mathcal{U}) \ .$$

Using (FO8.2) and the basic properties of the translation function $\kappa(\cdot)$ we can summarise these results as follows:

$$\kappa(\mathcal{T}_1 \cup \mathcal{T}_2) \diamond \kappa(\mathcal{U}) \models \kappa(\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{U}) \ .$$

But this contradicts (FO3) because although both $\mathcal{T}_1 \cup \mathcal{T}_2$ and $\mathcal{U}$ are consistent ontologies, $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{U}$ is inconsistent. □

[1] The function $\kappa(\cdot)$ is used to turn description logic axioms into their first-order representation. It was introduced in Definition 2.6.

An attentive reader might be worried about a small trick we used to establish this result. Namely, though we talk about TBox updates, the third assertion in $\mathcal{U}$ seems to "simulate" an ABox assertion. It may thus seem that undesirable behaviour of belief update operators occurs only if we either allow for ABox assertions, or the underlying DL is expressive enough to perform ABox assertions indirectly using TBox assertions, as was done in $\mathcal{U}$ using nominals. But note that the sole reason for having an ABox-like assertion in $\mathcal{U}$ is that in order for the proof to work, we need $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{U}$ to be inconsistent. Without it, we would ultimately not be able to prove that (FO3) is violated. We would nevertheless still conclude that $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{U}$ is *incoherent*, i.e. it contains an unsatisfiable concept name C, despite the fact that both $\mathcal{T}_1 \cup \mathcal{T}_2$ and $\mathcal{U}$ are coherent. Since coherence in this sense is one of the desirable properties for ontologies, we can maintain our result by eliminating the last assertion in $\mathcal{U}$ and replacing (FO3) with

(FO3') If both $\mathcal{T}$ and $\mathcal{U}$ are coherent, then $\mathcal{T} \diamond \mathcal{U}$ is also coherent.

Ultimately, Theorem 5.2 can be viewed from at least two different points of view. First, it is a *warning* for anyone considering to use a belief update operator for updating ontologies. Though there may be scenarios where this does not pose a problem, such as when the TBox is not present, in other scenarios this can be a major issue. This also means that even if belief updates were practically feasible for updating ontologies, they are by themselves insufficient. Note also that this essentially renders the hybrid update semantics suggested in Chapters 3 and 4 unsuitable for performing TBox updates. This is especially clear in case of the semantics from Chapters 3 because of the fact that the underlying first-order update operator satisfies postulate (FO8.2), and the very same postulate seems to be at the heart of the difficulties of model-based update operators when dealing with TBox updates.

From another viewpoint, Theorem 5.2 is a *challenge*. Where do things go wrong? And how can we fix them?

5.2 Clash Between Belief and Rule Updates

Regardless of the issues that belief updates have with TBoxes, as described in the previous section, they are still the main basis for updating ABoxes and thus play an important role in ontology updates. On the other hand, the rule update semantics presented in Section 2.8 were introduced as better alternatives to approaches based on belief change. It thus comes as no surprise that they differ fundamentally from belief change principles and operators.

Therefore, in order to devise an update semantics for hybrid knowledge bases that is faithful to both ontology and rule updates, it is necessary to somehow reconcile these two seemingly incompatible update paradigms. In the following we informally argue why such a reconciliation is difficult in the general case, i.e. when our goal is to define an update semantics that can deal with an arbitrary pair (or sequence) of hybrid knowledge bases.

The differences between belief and rule updates can be seen at multiple different levels. The most obvious ones are probably in the way they are defined. While belief update operators are characterised on the semantic level, by specifying the models of the result of an update, rule updates typically rely on the syntactic structure of underlying rules to determine the ones that need to be removed, rejected, or otherwise overridden by other rules. In case of causal rejection-based semantics, this rejection happens in context of a stable model candidate and a fixpoint condition is used to distinguish bad candidates from good ones. This means that the rejected rules and the resulting models are tightly interwoven: the model determines the rejected rules which in turn determine whether the model is any good.

The fixation of rule update semantics upon the syntactic structure of rules has two important consequences. First, since ontological axioms do not have the same syntactic structure as rules, i.e. there is no notion of a *body* and *head* of an ontological axiom, it is difficult to imagine how the ideas underlying rule updates could be applied to take care of updating ontologies. This essentially takes out the possibility of merely adapting a rule update semantics to deal with hybrid knowledge bases since the required syntactic structure is absent. The semantics that do not directly rely on bodies and heads of rules, such as RVS or RVD, are based on methods from belief revision which are also hard to combine with belief updates due to the different nature of these change operations and the incompatible principles that underlie them.

The second, more subtle consequence is that since rule update semantics respect support (c.f. Theorem 2.70), the rules provide justifications for literals, i.e. their activation is necessary to *sustain* the truth value of a literal. This is in stark contrast with belief updates where the actual axioms that brought about the truth of a literal are irrelevant and the literal then remains true by inertia regardless of whether the initial justification is still in effect. An example illustrating this behaviour follows:

Example 5.3 (Justifications)**.** Consider the programs

$$\mathcal{P}: \quad p \leftarrow q. \qquad\qquad \text{and} \qquad\qquad \mathcal{U}: \quad \neg q.$$
$$\qquad q.$$

Clearly, $\mathcal{P}$ has a single stable model in which p is true. However, for the DLP $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$ we obtain

$$[\![\boldsymbol{P}]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}]\!]_{\mathsf{DS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RD}} = [\![\boldsymbol{P}]\!]_{\mathsf{PRZ}} = [\![\boldsymbol{P}]\!]_{\mathsf{PRX_i}} = [\![\boldsymbol{P}]\!]_{\mathsf{RVS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RVD}} = \{\, \{\, \neg q \,\} \,\} \enspace .$$

In other words, p ceases to be true because the update removes its justification. By contrast, Winslett's operator would maintain the truth of p regardless of how the truth value of q changes. Particularly,

$$((q \supset p) \land q) \diamond_{\mathsf{W}} (\neg q) \models p \enspace .$$

Similarly, while a rule that has once been rejected can be reactivated in further updates, reinstating its head, this is not the case with belief updates.

Example 5.4 (Inertia)**.** Consider the program $\mathcal{P}$ from Example 5.3 and two updates

$$\mathcal{U}_1: \quad \neg p \leftarrow r. \qquad\qquad \text{and} \qquad\qquad \mathcal{U}_2: \quad \neg r.$$
$$\qquad r.$$

For the DLP $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U}_1, \mathcal{U}_2 \rangle$ we obtain

$$[\![\boldsymbol{P}]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}]\!]_{\mathsf{DS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RD}} = [\![\boldsymbol{P}]\!]_{\mathsf{PRX_i}} = [\![\boldsymbol{P}]\!]_{\mathsf{RVD}} = \{\, \{\, p, q, \neg r \,\} \,\} \enspace .$$

By contrast,

$$((q \supset p) \land q) \diamond_{\mathsf{W}} ((r \supset \neg p) \land r) \diamond_{\mathsf{W}} (\neg r) \models \neg p \enspace .$$

The essence of these differences is that while *inertia* is applied to *rules* in case of rule updates, Winslett's operator applies it to *literals*.

The above examples also point towards another important difference between belief and rule updates. The basic idea of all belief update operators, expressed in postulate (B8), is that each model represents a *possible world* that can be updated independently of other possible worlds. This would seem to roughly correspond with the fact that a logic program can have multiple stable models, each of them representing a different possible world. However, this parallel misses one important intuitive aspect of rule-based formalisms. Namely, while in belief updates the syntax of a belief base is used solely as a way of describing the set of plausible possible worlds, the rules in a program not only describe the stable models, they also encode essential relationships between literals which are not captured by any single stable model, nor by the set of all stable models. This is reflected in all rule update semantics, for instance in the property of *support* which puts syntactic conditions on the updated models. It should also be noted that this discussion seems to be related to the distinction between *coherence* and *foundational* theories for *belief revision* (Gärdenfors, 1990; Fuhrmann, 1991; Hansson, 1993b).

The following example demonstrates that the described differences between belief and rule updates easily clash with one another when we consider updates of simple hybrid knowledge bases.

Example 5.5 (Clash of Intuitions)**.** Consider the hybrid knowledge bases $\mathcal{K}_1 = (\mathcal{O}_1, \mathcal{P}_1)$ and $\mathcal{K}_2 = (\mathcal{O}_2, \mathcal{P}_2)$ where

$$\begin{aligned} \mathcal{O}_1 &: \quad q \supset p \\ \mathcal{P}_1 &: \quad q \leftarrow r. \end{aligned} \qquad \text{and} \qquad \begin{aligned} \mathcal{O}_2 &: \quad r \\ \mathcal{P}_2 &: \end{aligned}$$

What atoms should be true after an update of $\mathcal{K}_1$ by $\mathcal{K}_2$? Certainly, r must be true since it is directly asserted in $\mathcal{O}_2$. Following rule update semantics, we conclude that atom q must also be true because the rule in $\mathcal{P}_1$ is triggered by r and there is no reason to discard or override it. But it is unclear whether this should cause p to become true. On the one hand, it somehow seems that q is now true in the initial state, so in conjunction with the axiom in $\mathcal{O}_1$ we should conclude that p is true. On the other hand, q is justified by r which is only true in the updated state. Maybe it would be more correct to say that q is concluded in the updated state, so instead we should perform an update such as

$$(q \supset p) \diamond_{\mathrm{W}} q$$

In this case, we would not be able to conclude that p is true after the update.

What this example shows is a basic dilemma that arises due to the different mechanisms that bring about truth of literals under the distinct update paradigms. The tricky issue is not so much this particular dilemma, but the consequences it has on the general case where predicates can be defined in both the ontology and rules. How do we then keep track of whether a particular literal should remain justified by an active rule, or literal inertia should apply to it instead?

This fundamental question is supplemented by other confusing matters. For example, belief updates never recover from an inconsistent state – if initially there is no model, then this state is maintained regardless of the updates that we perform. Contrary to this, rule update semantics allow conflicts to be resolved by suitable updates. For instance, for the DLP

$$\boldsymbol{P} = \langle \{ p., \neg p. \}, \{ p. \} \rangle$$

we obtain

$$[\![\boldsymbol{P}]\!]_{\mathsf{AS}} = [\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\boldsymbol{P}]\!]_{\mathsf{DS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RD}} = [\![\boldsymbol{P}]\!]_{\mathsf{PRX_i}} = [\![\boldsymbol{P}]\!]_{\mathsf{RVS}} = [\![\boldsymbol{P}]\!]_{\mathsf{RVD}} = \{\{p\}\} \enspace .$$

In other words, all defined rule update semantics, except for PRZ,[2] assign a model to an inconsistent program after the conflict in it has been resolved in the update. This leads to troubling situations in case of hybrid knowledge bases: What do we do when a conflict occurs between a part of the ontology and some rules?

Overall, we believe that finding a universal hybrid update semantics poses many interrelated questions that need to be addressed incrementally, focusing on one aspect of the problem at a time. The essence of the described difficulties seems to stem from the syntactic foundations of rule updates which, on the one hand, provide the necessary expressivity to deal with literal dependencies stated in rules, but, on the other hand, prevent us from viewing both belief and rule updates from a unifying perspective. Since the syntactic structure and directionality of rules cannot be imposed on ontology axioms, bringing these two areas of research closer together amounts to finding a *semantic characterisation of rule updates* that retains their essential properties, such as *support*, and at the same time enables a more direct comparison with belief updates.

5.3 Approaching the Problems

We have shown in the previous sections that when considering updates of hybrid knowledge bases, one encounters a variety of problems. First, state-of-the-art ontology update and rule update semantics are both subject of active research, with many problems waiting to be collectively solved in appropriate ways. We have shown that TBox updates are a particular topic that has not been addressed much in the literature and for which model-based belief update operators are inappropriate. On the other hand, the TBox update operator proposed in (Calvanese et al., 2010) is based on ideas from belief *revision* and, in our opinion, lacks the flexibility to suitably capture the wide variety of use cases and TBox updates that will occur in them.

The second group of problems arises from attempting to combine existing approaches on ontology and rule updates in a common semantic framework. Due to the fundamental differences between these two update paradigms, this introduces a number of difficult questions regarding how the unified semantics should behave and how it can be defined.

In the following chapters we thus aim to clarify the relationship between belief and rule updates by searching for *semantic characterisations of rule updates*. If successful, this line of research can bring ontology and rule updates closer together and bring new insights into both of these separate research areas.

[2]The PRZ-semantics assigns no model to $\boldsymbol{P}$ because the initial program has no stable model. However, a consequence of this behaviour is that there are no PRZ-models even in situations when a model is typically expected. For instance, when $\mathcal{P} \cup \mathcal{U}$ has a stable model, this stable model should intuitively be assigned to the DLP $\langle \mathcal{P}, \mathcal{U} \rangle$. A particular example of this is when $\mathcal{P} = \{ p \leftarrow \sim p. \}$ and $\mathcal{U} = \{ p. \}$. Even though $\mathcal{P} \cup \mathcal{U}$ has a stable model, there is no PRZ-model of $\langle \mathcal{P}, \mathcal{U} \rangle$ because $\mathcal{P}$ has no stable model.

Part III

Semantic Rule Updates

6

Belief Updates on SE-Models

MARTIN: Yes! I have it! I have the representation theorem!

JOÃO: I'm not sure what you are talking about but it sounds good!

MARTIN: The update operators on SE-models? They work! I have a counterpart of KM postulates and a representation theorem!

JOÃO: And how are these operators? Behaving well?

MARTIN: I still haven't come up with a particular operator. I'll tell you as soon as I have one.

JANUARY 2010, LISBON, PORTUGAL

In this chapter we initiate our efforts to define rule update operators that do not rely on rule syntax and relate them to the traditional syntax-based rule update semantics. Our ultimate goal in this respect is to find a unifying perspective that would embrace both ontology and rule updates, enabling a deeper understanding of all involved methods and principles, creating room for their cross-fertilisation and so facilitating the search for a general update semantics for hybrid knowledge bases.

In Section 2.8 we showed that state-of-the-art rule update semantics are based on fundamentally different principles and methods than belief updates. Particularly, modifications on the level of individual stable models (Alferes and Pereira, 1996), akin to the belief update framework of Katsuno and Mendelzon (1991), are unable to capture the essential relationships between literals encoded in rules (Leite and Pereira, 1997).

Quite recently, AGM revision was successfully reformulated in the context of Logic Programming and specific revision operators for programs were investigated by Delgrande et al. (2008). Central to this novel approach are *SE-models* (Turner, 2003), a monotonic characterisation of logic programs that is strictly more expressive than stable models. The results of (Delgrande et al., 2008) constitute an important breakthrough in the research of logic program evolution because they change the focus from the syntactic representation of a program to its semantic content.

In this chapter we follow a similar path, but to tackle the problem of logic program *updates* instead of *revision* as in (Delgrande et al., 2008).

Using *SE-models*, we adapt the belief update postulates to rule updates and prove a representation theorem that provides a constructive characterisation of all rule update operators satisfying the postulates. This essentially makes it possible to define and evaluate any operator satisfying the postulates using an intuitive construction. We also define a concrete rule update operator that can be seen as a counterpart of Winslett's belief update operator (Winslett, 1990, see Section 2.5 for its definition).

However, while investigating the operator's properties, we uncover a serious drawback which, as it turns out, extends to all rule update operators based on SE-models and Katsuno and Mendelzon's approach to updates. In particular, it turns out that these operators are incompatible with the properties of *support* and *fact update* which are at the core of rule updates (c.f. Theorems 2.70 and 2.72). This is a very important finding as it guides the research on rule updates away from the semantic approach materialised in AGM and KM postulates or, alternatively, to the development of semantic characterisations of programs, richer than SE-models, that are appropriate for describing their dynamic behaviour.

The remainder of this chapter is structured as follows: In Section 6.1 we formally introduce SE-models and other required concepts and notation. Section 6.2 contains the reformulation of belief update postulates for rule updates together with a representation theorem and definition of a specific rule operator that satisfies the postulates. In Section 6.3 we analyse the previously defined operator and establish that all semantic rule update operators based on SE-models exhibit an undesired behaviour. We then summarise our findings in Section 6.4.

The relevant proofs are provided in Appendix D. A preliminary version of this work has been published in (Slota and Leite, 2010c).

6.1 Preliminaries

In this and the following chapters we constrain ourselves to propositional logic programs without explicit negation over a finite set of propositional atoms $\mathcal{A}$. The definitions of logic program syntax and semantics from Section 2.3 can be directly adapted to this constrained case. Unless stated otherwise, we do allow for disjunctive rules and for default literals in heads of rules. When referring to logic programs we frequently drop the designation "logic" since we no longer consider other types of programs, such as MKNF programs in Part II.

Without explicit negation, an ASP interpretation is simply a subset of $\mathcal{A}$ and coincides with the concept of a propositional interpretation. In the following we simply refer to both these concepts as *interpretations* and denote the set of all interpretations by $\mathcal{I} = 2^{\mathcal{A}}$.

SE-models (Turner, 2003), based on the non-classical logic of Here-and-There (Heyting, 1930; Pearce, 1997), provide a monotonic characterisation of logic programs that is expressive enough to capture both classical and stable models of a logic program. They can be viewed as three-valued interpretations.

Definition 6.1 (Three-Valued Interpretation)**.** A *three-valued interpretation* is a pair of interpretations $X = (I, J)$ such that $I \subseteq J$. Each atom $p \in \mathcal{A}$ is assigned one of three truth values in X:

$$X(p) = \begin{cases} \mathsf{T} & \text{if } p \in I; \\ \mathsf{U} & \text{if } p \in J \setminus I; \\ \mathsf{F} & \text{if } p \in \mathcal{A} \setminus J. \end{cases}$$

The set of all three-valued interpretations is denoted by $\mathfrak{X}$.

SE-models themselves are defined by referring to the same *program reduct* that we used to define stable models in Section 2.3.

Definition 6.2 (SE-Model)**.** Let $\mathcal{P}$ be a program. A three-valued interpretation (I, J) is an *SE-model of* $\mathcal{P}$ if $J \models \mathcal{P}$ and $I \models \mathcal{P}^J$. The set of all SE-models of $\mathcal{P}$ is denoted by $[\![\mathcal{P}]\!]_{\mathsf{SE}}$.

Note that $J \models \mathcal{P}$ if and only if $(J, J) \in [\![\mathcal{P}]\!]_{\mathsf{SE}}$, so SE-models capture the classical models of a program. And just like classical models, the set of SE-models of a program is monotonic, i.e. larger programs have smaller sets of SE-models. This is one of the important differences between SE-models and the non-monotonic stable models. Nevertheless, stable models of a program can be extracted from its set of SE-models.

Proposition 6.3 (Stable Models from SE-Models (Turner, 2003))**.** *An interpretation J is a stable model of a program $\mathcal{P}$ if and only if $(J, J) \in [\![\mathcal{P}]\!]_{\mathsf{SE}}$ and for all $I \subsetneq J$, $(I, J) \notin [\![\mathcal{P}]\!]_{\mathsf{SE}}$.*

This implies that programs with the same set of SE-models also have the same set of stable models. Moreover, when such programs are augmented with the same set of rules, the resulting programs still have the same set of stable models. In many situations such a property is very desirable as it allows one program to be modularly replaced by another one, even in the presence of additional rules, without affecting the resulting stable models. It is typically referred to as *strong equivalence* (Lifschitz et al., 2001).

Proposition 6.4 (SE-Models and Strong Equivalence (Turner, 2003))**.** *Let $\mathcal{P}$, $\mathcal{Q}$ be programs. Then $[\![\mathcal{P}]\!]_{\mathsf{SE}} = [\![\mathcal{Q}]\!]_{\mathsf{SE}}$ if and only if for every program $\mathcal{R}$, $[\![\mathcal{P} \cup \mathcal{R}]\!]_{\mathsf{SM}} = [\![\mathcal{Q} \cup \mathcal{R}]\!]_{\mathsf{SM}}$.*

This property also explains the origin of the name *SE-models* – SE stands for *strong equivalence*. Therefore, we define strong equivalence and entailment as follows:

Definition 6.5 (Strong Equivalence and Strong Entailment)**.** Let $\mathcal{P}$, $\mathcal{Q}$ be programs. We say that $\mathcal{P}$ *strongly entails* $\mathcal{Q}$, denoted by $\mathcal{P} \models_{\mathsf{SE}} \mathcal{Q}$, if $[\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{SE}}$, and that $\mathcal{P}$ *is strongly equivalent to* $\mathcal{Q}$, denoted by $\mathcal{P} \equiv_{\mathsf{SE}} \mathcal{Q}$, if $[\![\mathcal{P}]\!]_{\mathsf{SE}} = [\![\mathcal{Q}]\!]_{\mathsf{SE}}$.

One distinguishing property of SE-models that we will need to carefully consider in the following sections is that whenever a program $\mathcal{P}$ has the SE-model (I, J), it also has the SE-model (J, J). In other words, whenever $I \subsetneq J$, there is no program that has the single SE-model (I, J). Delgrande et al. (2008) thus say that a set of SE-models of a program is always *well-defined*. We introduce the following concepts in order to be able to talk about this property:

Definition 6.6 (Well-Defined Set of Interpretations and Basic Program)**.** A set of three-valued interpretations $\mathcal{M}$ is *well-defined* if for every three-valued interpretation (I, J), $(I, J) \in \mathcal{M}$ implies $(J, J) \in \mathcal{M}$.

For every three-valued interpretation $X = (I, J)$ we denote by X^* the three-valued interpretation (J, J). We say that a program $\mathcal{P}$ is *basic* if $[\![\mathcal{P}]\!]_{\mathsf{SE}} = \{ X, X^* \}$ for some three-valued interpretation X.

A program is basic if it either has a unique SE-model (J, J), or a pair of SE-models (I, J) and (J, J). In the former case, the program exactly determines the truth values of all literals – the atoms in J are true and the atoms in $\mathcal{A} \setminus J$ are false. In the latter case, the program only forces the atoms in I to be true and the atoms in $J \setminus I$ may either be undefined or true, as long as they all have the same truth value.

The following proposition formally pinpoints the fact that the set of SE-models of any program is well-defined and, vice versa, every well-defined set of three-valued interpretations is the set of SE-models of some program.

Proposition 6.7 (Delgrande et al. (2008))**.** *A set of three-valued interpretations $\mathcal{M}$ is well-defined if and only if $\mathcal{M} = [\![\mathcal{P}]\!]_{\mathsf{SE}}$ for some program $\mathcal{P}$.*

6.2 Semantic Rule Update Operators Based on SE-Models

With the necessary concepts defined, we are ready to step forward and tailor the belief update postulates and operators to the context of logic programs and SE-models. Similarly as in the case of belief updates, we liberally define a rule update operator as any function that takes two inputs, the original program and its update, and returns the updated program.

Definition 6.8 (Rule Update Operator)**.** A *rule update operator* is a binary function on the set of all programs.

In order to reformulate postulates (B1) – (B8) for programs under the SE-models semantics, we first need to specify what a conjunction and disjunction of logic programs is. To this end, we introduce program conjunction and disjunction operators. These are required to assign, to each pair of programs, a program whose set of SE-models is the intersection and union, respectively, of the sets of SE-models of argument programs.

Definition 6.9 (Program Conjunction and Disjunction)**.** A binary operator $\dot{\wedge}$ on the set of all programs is a *program conjunction operator* if for all programs $\mathcal{P}$, $\mathcal{Q}$,

$$[\![\mathcal{P} \mathbin{\dot{\wedge}} \mathcal{Q}]\!]_{\mathsf{SE}} = [\![\mathcal{P}]\!]_{\mathsf{SE}} \cap [\![\mathcal{Q}]\!]_{\mathsf{SE}} \ .$$

A binary operator $\dot{\vee}$ on the set of all programs is a *program disjunction operator* if for all programs $\mathcal{P}$, $\mathcal{Q}$,

$$[\![\mathcal{P} \mathbin{\dot{\vee}} \mathcal{Q}]\!]_{\mathsf{SE}} = [\![\mathcal{P}]\!]_{\mathsf{SE}} \cup [\![\mathcal{Q}]\!]_{\mathsf{SE}} \ .$$

In the following we assume that some program conjunction and disjunction operators $\dot{\wedge}$, $\dot{\vee}$ are given. Note that the program conjunction operator may simply return the union of argument programs; it is the same as the *expansion operator* defined in (Delgrande et al., 2008). A program disjunction operator can be defined by translating the argument programs into the logic of Here-and-There (Heyting, 1930; Łukasiewicz, 1941; Pearce, 1997), taking their disjunction and transforming the resulting formula back into a logic program (using results from (Cabalar and Ferraris, 2007)).

The final obstacle before we can proceed with introducing the new postulates is the following: We need to substitute the notion of a *complete formula* used in (B7) with a suitable class of logic programs. It turns out that the notion of a *basic program*, as introduced in Definition 6.6, is a natural candidate for this purpose. While a complete formula is defined as having a unique model, a program is basic if it has either a unique SE-model (J, J), or a pair of SE-models (I, J) and (J, J). The latter case needs to be allowed in order to make the new postulate applicable to three-valued interpretations (I, J) with $I \subsetneq J$ because no program has the single SE-model (I, J).

The following are the reformulated postulates for a rule update operator $\oplus$ and programs $\mathcal{P}$, $\mathcal{Q}, \mathcal{U}, \mathcal{V}$:

(P1)$_{\mathsf{SE}}$ $\mathcal{P} \oplus \mathcal{U} \models_{\mathsf{SE}} \mathcal{U}$.

(P2)$_{\mathsf{SE}}$ If $\mathcal{P} \models_{\mathsf{SE}} \mathcal{U}$, then $\mathcal{P} \oplus \mathcal{U} \equiv_{\mathsf{SE}} \mathcal{P}$.

(P3)$_{\mathsf{SE}}$ If $[\![\mathcal{P}]\!]_{\mathsf{SE}} \neq \emptyset$ and $[\![\mathcal{U}]\!]_{\mathsf{SE}} \neq \emptyset$, then $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} \neq \emptyset$.

(P4)$_{\mathsf{SE}}$ If $\mathcal{P} \equiv_{\mathsf{SE}} \mathcal{Q}$ and $\mathcal{U} \equiv_{\mathsf{SE}} \mathcal{V}$, then $\mathcal{P} \oplus \mathcal{U} \equiv_{\mathsf{SE}} \mathcal{Q} \oplus \mathcal{V}$.

(P5)$_{\mathsf{SE}}$ $(\mathcal{P} \oplus \mathcal{U}) \mathbin{\dot\wedge} \mathcal{V} \models_{\mathsf{SE}} \mathcal{P} \oplus (\mathcal{U} \mathbin{\dot\wedge} \mathcal{V})$.

(P6)$_{\mathsf{SE}}$ If $\mathcal{P} \oplus \mathcal{U} \models_{\mathsf{SE}} \mathcal{V}$ and $\mathcal{P} \oplus \mathcal{V} \models_{\mathsf{SE}} \mathcal{U}$, then $\mathcal{P} \oplus \mathcal{U} \equiv_{\mathsf{SE}} \mathcal{P} \oplus \mathcal{V}$.

(P7)$_{\mathsf{SE}}$ If $\mathcal{P}$ is basic, then $(\mathcal{P} \oplus \mathcal{U}) \mathbin{\dot\wedge} (\mathcal{P} \oplus \mathcal{V}) \models_{\mathsf{SE}} \mathcal{P} \oplus (\mathcal{U} \mathbin{\dot\vee} \mathcal{V})$.

(P8)$_{\mathsf{SE}}$ $(\mathcal{P} \mathbin{\dot\vee} \mathcal{Q}) \oplus \mathcal{U} \equiv_{\mathsf{SE}} (\mathcal{P} \oplus \mathcal{U}) \mathbin{\dot\vee} (\mathcal{Q} \oplus \mathcal{U})$.

We can also formulate the weakened versions of postulates (P2)$_{\mathsf{SE}}$ and (P4)$_{\mathsf{SE}}$, useful when we need to pinpoint a particular part of a postulate that causes a certain behaviour.

(P2.⊤)$_{\mathsf{SE}}$ $\mathcal{P} \oplus \emptyset \equiv_{\mathsf{SE}} \mathcal{P}$.

(P2.1)$_{\mathsf{SE}}$ $\mathcal{P} \mathbin{\dot\wedge} \mathcal{U} \models_{\mathsf{SE}} \mathcal{P} \oplus \mathcal{U}$.

(P2.2)$_{\mathsf{SE}}$ $(\mathcal{P} \mathbin{\dot\wedge} \mathcal{U}) \oplus \mathcal{U} \models_{\mathsf{SE}} \mathcal{P}$.

(P4.1)$_{\mathsf{SE}}$ If $\mathcal{P} \equiv_{\mathsf{SE}} \mathcal{Q}$, then $\mathcal{P} \oplus \mathcal{U} \equiv_{\mathsf{SE}} \mathcal{Q} \oplus \mathcal{U}$.

(P4.2)$_{\mathsf{SE}}$ If $\mathcal{U} \equiv_{\mathsf{SE}} \mathcal{V}$, then $\mathcal{P} \oplus \mathcal{U} \equiv_{\mathsf{SE}} \mathcal{P} \oplus \mathcal{V}$.

We now turn to a constructive characterisation of rule update operators satisfying conditions (P1)$_{\mathsf{SE}}$ – (P8)$_{\mathsf{SE}}$. Analogically to belief updates, it is based on an order assignment, but this time over the set of all three-valued interpretations $\mathcal{X}$. Since the set of SE-models of a program must be well-defined, not every order assignment characterises a rule update operator. We thus additionally define *well-defined order assignments* as those that do.

Definition 6.10 (Rule Update Operator Characterised by an Order Assignment)**.** Let $\oplus$ be a rule update operator and ω a preorder assignment over $\mathcal{X}$. We say that $\oplus$ is *characterised by* ω if for all programs $\mathcal{P}, \mathcal{U}$,

$$[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq^X_\omega\right) \quad .$$

We say that a preorder assignment over $\mathcal{X}$ is *well-defined* if some rule update operator is characterised by it.

Similarly as with belief updates, we require the order assignment to be faithful, i.e. to consider each three-valued interpretation the closest to itself.

Definition 6.11 (Faithful Order Assignment)**.** A preorder assignment ω over $\mathcal{X}$ is *faithful* if for every three-valued interpretation X the following condition is satisfied:

$$\text{For every } Y \in \mathcal{X} \text{ with } Y \neq X \text{ it holds that } X <^X_\omega Y \quad .$$

Interestingly, faithful assignments characterise the same class of operators as the larger class of semi-faithful assignments, defined as follows:

Definition 6.12 (Semi-Faithful Order Assignment)**.** A preorder assignment ω over $\mathcal{X}$ is *semi-faithful* if for every three-valued interpretation X the following conditions are satisfied:

1. For every $Y \in \mathcal{X}$ with $Y \neq X$ and $Y \neq X^*$, either $X <^X_\omega Y$ or $X^* <^X_\omega Y$.
2. If $X^* \leq^X_\omega X$, then $X \leq^X_\omega X^*$.

Finally, we require the preorder assignment to satisfy one further condition, related to the well-definedness of sets of SE-models of every program. It can naturally be seen as the semantic counterpart of $(P7)_{SE}$.

Definition 6.13 (Organised Preorder Assignment)**.** A preorder assignment ω is *organised* if for all three-valued interpretations X, Y and all well-defined sets of three-valued interpretations $\mathcal{M}, \mathcal{N}$ the following condition is satisfied:

$$\text{If } Y \in \min(\mathcal{M}, \leq^X_\omega) \cup \min(\mathcal{M}, \leq^{X^*}_\omega) \text{ and } Y \in \min(\mathcal{N}, \leq^X_\omega) \cup \min(\mathcal{N}, \leq^{X^*}_\omega),$$
$$\text{then } Y \in \min(\mathcal{M} \cup \mathcal{N}, \leq^X_\omega) \cup \min(\mathcal{M} \cup \mathcal{N}, \leq^{X^*}_\omega).$$

We are now ready to formulate the main result of this section:

Theorem 6.14 (Representation Theorem)**.** *Let $\oplus$ be a rule update operator. The following conditions are equivalent:*

a) The operator $\oplus$ satisfies conditions $(P1)_{SE}$ – $(P8)_{SE}$.

b) The operator $\oplus$ is characterised by a semi-faithful and organised preorder assignment.

c) The operator $\oplus$ is characterised by a faithful and organised partial order assignment.

Proof. See Appendix D, page 236. □

This theorem provides a constructive characterisation of rule update operators satisfying the defined postulates. It facilitates the analysis of their properties, both semantic as well as computational. Note also that it implies that the larger class of *semi-faithful* and organised *preorder* assignments is equivalent to the smaller class of *faithful* and organised *partial order* assignments. Furthermore, it offers a strategy for defining operators satisfying the postulates that can be directly implemented. This strategy is also complete in the sense that, up to strong equivalence, all operators satisfying the postulates can be characterised and distinguished by applying this strategy.

In what follows, we define a specific update operator based on the ideas underlying Winslett's belief update operator (Winslett, 1990) defined in Section 2.5. Similarly as was argued in (Delgrande et al., 2008), since we are working with well-defined sets of three-valued interpretations, preference needs to be given to their second component. We define the assignment W for all three-valued interpretations $X = (I, J)$, $Y = (K_1, L_1)$, $Z = (K_2, L_2)$ as follows: $Y \leq^X_{\mathsf{W}} Z$ if and only if the following conditions are satisfied:

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

Intuitively, first we compare the differences between the second components of Y and Z w.r.t. X. If they are equal, we compare the differences between the first components of Y and Z w.r.t. X, but now ignoring the differences between the second components.

The following result shows that W indeed satisfies the necessary conditions to characterise rule update operators satisfying the postulates.

Proposition 6.15. *W is a well-defined, faithful and organised preorder assignment.*

Proof. See Appendix D, page 239. □

Since W is well-defined, there exist rule update operators characterised by it. Furthermore, as a consequence of Theorem 6.14 and Proposition 6.15:

Corollary 6.16. *Rule update operators characterised by* W *satisfy conditions* (P1)$_{SE}$ – (P8)$_{SE}$.

6.3 Syntactic Properties of Semantic Rule Updates

One of the benefits of dealing with rule updates on the semantic level is that semantic properties that are rather difficult to show for syntax-based update operators are much easier to analyse and prove. For example, one of the most widespread and counterintuitive side-effects of syntax-based rule update semantics is that they are sensitive to tautological updates. In case of semantic update operators, such a behaviour is impossible given that the operator satisfies (P2.⊤)$_{SE}$ and (P4.2)$_{SE}$.

However, semantic update operators do not always behave the way we expect. Consider first an example using some update operator $\oplus_{W}$ characterised by the order assignment W defined in the previous section:

Example 6.17. Suppose that $\mathcal{A} = \{ p, q \}$, i.e. there are exactly two propositional atoms, and let the programs $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{U}$ be as follows:

$$\mathcal{P}: \quad p. \qquad\qquad \mathcal{Q}: \quad p \leftarrow q. \qquad\qquad \mathcal{U}: \quad \sim q.$$
$$\qquad q. \qquad\qquad\qquad q.$$

It can be verified that:[1]

$$[\![\mathcal{P} \oplus_{W} \mathcal{U}]\!]_{SE} = [\![\mathcal{Q} \oplus_{W} \mathcal{U}]\!]_{SE} = \{ (p, p) \} \quad .$$

Hence both $\mathcal{P} \oplus_{W} \mathcal{U}$ and $\mathcal{Q} \oplus_{W} \mathcal{U}$ have the single stable model $J = \{ p \}$. In case of $\mathcal{P} \oplus_{W} \mathcal{U}$ this is indeed the expected result. But in case of $\mathcal{Q} \oplus_{W} \mathcal{U}$ we can see that p is true in J even though there is no rule in $\mathcal{Q} \cup \mathcal{U}$ justifying it, i.e. there is no rule with p in its head and its body satisfied in J. In other words, the rule update semantics induced by $\oplus_{W}$ *does not respect support*.

We have seen in Section 2.8 that support is one of the fundamental properties of rule update semantics – a wide range of semantics, from causal rejection-based ones, to ones based on ideas from belief revision, respect support (c.f. Theorem 2.70). As the above example shows, this is not the case with the rule update operator $\oplus_{W}$.

Unfortunately, the violation of fundamental properties of existing rule update semantics is not specific to the operator $\oplus_{W}$. The following theorem shows that every rule update operator satisfying (P4.1)$_{SE}$ violates either *support* or *fact update*, another basic property satisfied by a range of different rule update semantics (c.f. Theorem 2.72).

Theorem 6.18. *A rule update operator that satisfies* (P4.1)$_{SE}$ *either does not respect support or it does not respect fact update.*

Proof. Let $\oplus$ be a rule update operator that satisfies (P4.1)$_{SE}$ and consider again the programs $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{U}$ from Example 6.17. Since $\mathcal{P}$ is strongly equivalent to $\mathcal{Q}$, by (P4.1)$_{SE}$ we obtain that $\mathcal{P} \oplus \mathcal{U}$ is strongly equivalent to $\mathcal{Q} \oplus \mathcal{U}$. Consequently, $\mathcal{P} \oplus \mathcal{U}$ has the same stable

[1] For the sake of readability we omit the usual set notation when listing SE-interpretations. For example, instead of $(\{ p \}, \{ p, q \})$ we simply write (p, pq).

models as $\mathcal{Q} \oplus \mathcal{U}$. It only remains to observe that if $\oplus$ respects fact update, then $\mathcal{P} \oplus \mathcal{U}$ has the unique stable model $\{ p \}$. But then $\{ p \}$ is a stable model of $\mathcal{Q} \oplus \mathcal{U}$ in which p is unsupported by $\mathcal{Q} \cup \mathcal{U}$. Hence $\oplus$ does not respect support. □

This theorem shows that rule update operators based on SE-models and Katsuno and Mendelzon's belief update framework necessarily violate either support or fact update. These two properties are based on fundamental and widely accepted intuitions. They are by no means exhaustive or sufficient – it is not difficult to define rule update operators that respect both support and fact update but are sensitive to tautological updates or quickly end up in an inconsistent state without a possibility of recovery. But both properties seem necessary, even elementary properties of a well-behaved rule update operator.

Moreover, the principle $(\mathsf{P4})_{\mathsf{SE}}$ is also adopted for *revision* of logic programs based on SE-models in (Delgrande et al., 2008).[2] This means that Theorem 6.18 extends to semantic program *revision* operators, such as those defined in (Delgrande et al., 2008).

6.4 Discussion

The previous sections have revealed how belief update postulates and operators can be reformulated using SE-models to arrive at rule update operators with clear semantic characterisations. We have shown a rule update analogue of the representation theorem for belief updates and defined a specific rule update operator, akin to Winslett's belief update operator, that satisfies counterparts of belief update postulates. The semantic nature of update operators that fit within the defined update framework not only facilitates the investigation of their properties, both semantic as well as computational, but it also provides an intuitive strategy for constructively defining these operators. This is the first major contribution of this chapter since it brings, for the first time, updates of logic programs in line with belief update principles and methods.

The second important contribution is the uncovering of a serious drawback that extends to all rule update and revision operators based on SE-models and AGM-style approach to program revision and update. All such operators violate at least one of two fundamental albeit basic properties satisfied by a range of rule update semantics: respect for *support* and for *fact update*. This might be mitigated if a richer semantic characterisation of logic programs was used instead of SE-models. Such a characterisation would have to be able to distinguish between programs such as $\mathcal{P} = \{ p., q. \}$ and $\mathcal{Q} = \{ p \leftarrow q., q. \}$ because they are expected to behave differently when subject to evolution.

Thus, our findings are very important as they guide further research on updates of logic programs either

a) away from the semantic approach materialised in AGM-style postulates, or
b) to the development of semantic characterisations of logic programs that are richer than SE-models and appropriately capture their dynamic behaviour, or even
c) back to the syntax-based rule update semantics.

[2] Note that the belief update postulate (B4), from which $(\mathsf{P4})_{\mathsf{SE}}$ originates, is also one of the reformulated AGM postulates for belief *revision* (Katsuno and Mendelzon, 1992). The original AGM framework (Alchourrón et al., 1985) assumes that the initial knowledge base $\mathcal{B}$ is closed w.r.t. logical consequence and the first AGM postulate requires that the result of revision also be a closed set. Under these assumptions, different knowledge bases cannot be equivalent and, as a consequence, the original AGM postulate corresponding to (B4) is $\star 5$: If $Cn(\mu) = Cn(\nu)$, then $\mathcal{B} \star \mu = \mathcal{B} \star \nu$ (where Cn is the logical consequence operator and $\star$ the revision operator).

The following two chapters combine the latter two options. First, in Chapter 7, we study the possibility of viewing a program as the *set of sets of models of its rules*. We consider SE-models as well as a novel semantic characterisation of rules for this purpose and study the associated notions of program equivalence and entailment. The obtained results are vital in Chapter 8 where we propose to perform rule updates by introducing new models – *exceptions* – to the sets of models of rules in the original program. This leads us towards a semantic characterisation of the historically first rule update semantics, bridging the original syntax-based approach to rule updates with a semantic one.

7

Semantic Characterisations of Rules and Programs

JOÃO: I've had this idea during the talks today, let me tell you before I forget. So we know that SE-models of a program are insufficient to perform updates. How about if we keep more information. . . like the sets of SE-models of all the rules?

MARTIN: I'm not sure I'm following.

JOÃO: I mean that we could look at a program as the set of sets of SE-models of its rules. Could we then perform updates better?

MARTIN: I'm not sure. . . maybe. I'll think about it.

MAY 2010, TORONTO, CANADA

We have established in the previous chapter that updating a logic program through its set of SE-models is insufficient if we want to adhere to desirable properties of rule updates, namely *support* and *fact update*. Since our main objective is the study of semantic approaches to rule updates and their relationship with syntax-based ones, we need to look for richer semantic characterisations of programs that allow for update operators that are in line with properties such as support and fact update.

In the present chapter we focus on the possibility of viewing a logic program as the *set of sets of models of its rules*. This way we acknowledge rules as the atomic pieces of knowledge in a program and can distinguish many more programs than by considering only the overall set of models of the program. As we shall see in Chapter 8, we can then devise rule update operators that not only respect both support and fact update, but provide a direct semantic characterisation of the historically first syntax-based approach to rule updates: the JU-semantics (for a definition see Section 2.8, page 44).

The apparent question that remains is: What kind of models do we choose for the individual rules in a program? SE-models seem to be a natural choice because stable models of a program can be inferred from its set of SE-models, which in turn can be obtained by intersecting the sets of SE-models of its rules. This brings out the question of the expressivity of SE-models w.r.t. a single rule: Which rules can be distinguished using SE-models and which ones cannot?

We provide an answer to this question in Section 7.1 by summarising and extending the results about SE-models from the literature. Our main finding is that although SE-models conveniently filter out many insignificant differences between rules, they are also unable to distinguish some rules which are treated differently by many rule update semantics. For example, rules such as $(\sim p \leftarrow q.)$ and $(\sim q \leftarrow p.)$ have the same SE-models, making it impossible to use SE-models to emulate rule update semantics that rely on distinguishing the literals in rule heads.

In order to enable such rules to be distinguished semantically, in Section 7.2 we propose a novel characterisation of rules and programs which we dub *RE-models*. We show that RE-models retain the essential properties of SE-models and so form a suitable basis for characterising rule updates.

Subsequently, in Section 7.3, we review notions of program equivalence and entailment known from the literature and compare them to those that naturally follow from viewing a program as a set of sets of SE- or RE-models.

The relevant proofs can be found in Appendix E. Preliminary versions of some parts of this chapter have appeared in (Slota and Leite, 2011, 2012a).

7.1 SE-Models vs. Individual Rules

The objective of this section is to understand the relationship between SE-models and single rules. We continue to work under the assumptions introduced in Chapter 6: we consider propositional logic programs without explicit negation over a finite set of propositional atoms $\mathcal{A}$. For more details and a definition of SE-models of logic programs we refer the reader to Section 6.1.

We start by introducing the non-standard notion of a *canonical tautology*, a unique rule that represents the class of rules satisfied in any interpretation. It will be useful throughout the remainder of this thesis.

Definition 7.1 (Canonical Tautology)**.** Let p_τ be a fixed atom from $\mathcal{A}$. The *canonical tautology* τ is the rule $(p_\tau \leftarrow p_\tau.)$.

SE-models of a single rule are defined analogically to SE-models of logic programs, using the notion of a *rule reduct*.

Definition 7.2 (Reduct and SE-Model of a Rule)**.** Let π be a rule and J an interpretation. The *reduct of π relative to J* is the rule

$$\pi^J = \begin{cases} \tau & \text{if } J \models (\sim H(\pi)^- \leftarrow \sim B(\pi)^-.) \\ H(\pi)^+ \leftarrow B(\pi)^+. & \text{otherwise} \end{cases}$$

A three-valued interpretation (I, J) is an *SE-model of* π if $J \models \pi$ and $I \models \pi^J$. The set of all SE-models of a rule π is denoted by $[\![\pi]\!]_{\mathsf{SE}}$.

We say that π is *(SE-)tautological* if $[\![\pi]\!]_{\mathsf{SE}} = \mathcal{X}$. Rules π, σ are *SE-equivalent* if $[\![\pi]\!]_{\mathsf{SE}} = [\![\sigma]\!]_{\mathsf{SE}}$. A set of three-valued interpretations $\mathcal{M}$ is *SE-rule-expressible* if $\mathcal{M} = [\![\pi]\!]_{\mathsf{SE}}$ for some rule π.

Note that the canonical tautology is SE-tautological. Also, given a program $\mathcal{P}$, it is not difficult to verify that $[\![\mathcal{P}]\!]_{\mathsf{SE}} = \bigcap_{\pi \in \mathcal{P}} [\![\pi]\!]_{\mathsf{SE}}$.

In order to formally pinpoint the expressivity of SE-models w.r.t. individual rules, in Section 7.1.1 we introduce a set of representatives of rule equivalence classes induced by SE-models and show how the representative of a class can be constructed given one of its members. Subsequently, Section 7.1.2 shows how to reconstruct a representative from the set of its SE-models. In Section 7.1.3 we determine the conditions under which a set of three-valued interpretations is SE-rule-expressible.

7.1.1 SE-Canonical Rules

We start by bringing out simple but powerful transformations that simplify a given rule while preserving its SE-models. Although many of these results have already been formulated in various ways before (Inoue and Sakama, 1998; Janhunen, 2001; Inoue and Sakama, 2004; Cabalar et al., 2007), here we present the ones relevant to identify a set of representatives of rule equivalence classes induced by SE-models.

For the rest of this section we assume that H and B are sets of literals, p is an atom and L a literal. The following result summarises the conditions under which a rule is tautological:

Lemma 7.3 (Consequence of Theorem 4.4 in (Inoue and Sakama, 2004); part i) of Lemma 2 in (Cabalar et al., 2007)). *Rules of the following forms are SE-tautological:*

$$p; H \leftarrow p, B. \qquad H; {\sim}p \leftarrow B, {\sim}p. \qquad H \leftarrow B, p, {\sim}p.$$

Proof. See Appendix E, page 242. □

Lemma 7.3 shows that repeating an atom in different "components" of the rule frequently causes the rule to be tautological. In particular, this happens if the same atom occurs in the positive head and positive body, or in the negative head and negative body, or in the positive and negative bodies of a rule. How about the cases when the head contains the negation of a literal from the body? The following Lemma clarifies this situation:

Lemma 7.4 (Consequence of (3) and (4) in Lemma 1 in (Cabalar et al., 2007)). *Rules of the following forms are SE-equivalent:*

$$H; {\sim}L \leftarrow L, B. \qquad H \leftarrow L, B.$$

Proof. See Appendix E, page 243. □

So if a literal is present in the body of a rule, its negation can be removed from the head without affecting the SE-models of the rule.

Until now we have seen that a rule π that has a common atom in at least two of the sets $H(\pi)^+ \cup H(\pi)^-$, $B(\pi)^+$ and $B(\pi)^-$ is either tautological, or SE-equivalent to a rule where the atom is omitted from the rule's head. In other words, π is SE-equivalent either to the canonical tautology τ, or to a rule without such repetitions.

Perhaps surprisingly, repetitions in the positive and negative heads cannot be simplified away. For example, over the alphabet $\mathcal{A}_p = \{ p \}$, the disjunctive rule $(p; {\sim}p \leftarrow .)$ has two SE-models, $(\emptyset, \emptyset)$ and $(\{ p \}, \{ p \})$, so it is not SE-tautological, nor is it SE-equivalent to any of the facts $(p.)$ and $({\sim}p.)$. Actually, it is not very difficult to see that it is not SE-equivalent to *any* other rule, even over larger alphabets. So the fact that an atom is in both $H(\pi)^+$ and $H(\pi)^-$ cannot all by itself imply that some kind of SE-models preserving rule simplification is possible.

The final result reveals a special case in which we can eliminate the whole negative head of a rule and move it to its positive body. This occurs whenever the positive head is empty.

Lemma 7.5 (Related to Corollary 4.10 in (Inoue and Sakama, 1998) and to Corollary 1 in (Cabalar et al., 2007))**.** *Rules of the following forms are SE-equivalent:*

$$\sim p; \sim H^- \leftarrow B. \qquad\qquad \sim H^- \leftarrow p, B.$$

Proof. See Appendix E, page 243. □

Armed with the above results, we can introduce the notion of an SE-canonical rule. Each such rule represents a different rule equivalence class induced by the SE-models semantics. In other words, every rule is SE-equivalent to exactly one SE-canonical rule. After the definition, we provide constructive transformations which show that this is indeed the case. Note that the definition can be derived directly from the lemmas above:

Definition 7.6 (SE-Canonical Rule)**.** We say that a rule π is *SE-canonical* if either it is τ, or the following conditions are satisfied:

1. The sets $H(\pi)^+ \cup H(\pi)^-$, $B(\pi)^+$ and $B(\pi)^-$ are pairwise disjoint.
2. If $H(\pi)^+$ is empty, then $H(\pi)^-$ is also empty.

This definition is closely related to the notion of a *fundamental rule* introduced in Definition 1 of (Cabalar et al., 2007). There are two differences between SE-canonical and fundamental rules:

(1) A fundamental rule need not satisfy condition 2. above;

(2) No SE-tautological rule is fundamental.

As a consequence, fundamental rules do not cover all SE-rule-expressible sets of three-valued interpretations, and two distinct fundamental rules may still be SE-equivalent. From the point of view of rule equivalence classes induced by the SE-models semantics, there is one class that contains no fundamental rule (the class of SE-tautological rules), and some classes contain more than one fundamental rule. In the following we show that SE-canonical rules overcome both limitations of fundamental rules, i.e. every rule is SE-equivalent to exactly one SE-canonical rule. To this end, we define constructive transformations that directly show the mutual relations between rule syntax and semantics.

The following transformation provides a direct way of constructing an SE-canonical rule that is SE-equivalent to a given rule π.

Definition 7.7 (Transformation into an SE-Canonical Rule)**.** Given a rule π, we define the SE-canonical rule $\mathsf{can}_{\mathsf{SE}}(\pi)$ as follows:

(i) If any of the sets $H(\pi)^+ \cap B(\pi)^+$, $H(\pi)^- \cap B(\pi)^-$ and $B(\pi)^+ \cap B(\pi)^-$ is non-empty, then $\mathsf{can}_{\mathsf{SE}}(\pi) = \tau$.

(ii) If (i) does not apply and $H(\pi)^+ \setminus B(\pi)^- \neq \emptyset$, then $\mathsf{can}_{\mathsf{SE}}(\pi)$ is the rule

$$(H(\pi)^+ \setminus B(\pi)^-); \sim(H(\pi)^- \setminus B(\pi)^+) \leftarrow B(\pi)^+, \sim B(\pi)^-.$$

(iii) If (i) does not apply and $H(\pi)^+ \setminus B(\pi)^- = \emptyset$, then $\mathsf{can}_{\mathsf{SE}}(\pi)$ is the constraint

$$\leftarrow (B(\pi)^+ \cup H(\pi)^-), \sim B(\pi)^-.$$

Correctness of the transformation follows directly from Lemmas 7.3, 7.4 and 7.5.

Theorem 7.8. *Every rule π is SE-equivalent to the SE-canonical rule* $\mathsf{can}_{\mathsf{SE}}(\pi)$.

Proof. See Appendix E, page 244. □

What remains to be proven is that no two different SE-canonical rules are SE-equivalent. In the next section we show how every SE-canonical rule can be reconstructed from the set of its SE-models. As a consequence, no two different SE-canonical rules can have the same set of SE-models.

7.1.2 Reconstructing Rules from SE-Models

In order to reconstruct a rule π from the set of its SE-models, we need to understand how exactly each literal in the rule influences its models. The following lemma provides a useful characterisation of the set of countermodels of a rule in terms of its syntax:

Lemma 7.9 (Different formulation of Theorem 4 in (Cabalar et al., 2007))**.** *Let (I, J) be a three-valued interpretation and π a rule. Then $(I, J) \notin [\![\pi]\!]_{\mathsf{SE}}$ if and only if the following conditions are satisfied:*

1. *Either $B(\pi)^+ \subseteq I \subseteq \mathcal{A} \setminus H(\pi)^+$ or $J \cap H(\pi)^+ = \emptyset$ and*
2. *$H(\pi)^- \cup B(\pi)^+ \subseteq J \subseteq \mathcal{A} \setminus B(\pi)^-$.*

Proof. See Appendix E, page 244. □

If we take a closer look at the conditions in Lemma 7.9, we find that the presence of an atom from $B(\pi)^-$ in J guarantees that the second condition is falsified, so (I, J) is an SE-model of π, regardless of the content of I. Somewhat similar is the situation with positive head atoms – whenever an atom from $H(\pi)^+$ is present in I, the first condition is falsified and (I, J) is an SE-model of π. More formally, given a rule π, for every atom $p \in B(\pi)^-$ it holds that

$$p \in J \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \tag{7.1}$$

and for every atom $p \in H(\pi)^+$ it holds that

$$p \in I \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \ . \tag{7.2}$$

If we restrict ourselves to SE-canonical rules different from τ, we find that these conditions are not only necessary, but, when combined properly, also sufficient to decide what atoms belong to the negative body and positive head of π.

If not stated otherwise, we assume in the rest of this section that π is an SE-canonical rule different from τ. Keeping in mind that every atom that satisfies condition (7.1) also satisfies condition (7.2) (because I is a subset of J), and that $B(\pi)^-$ is by definition of an SE-canonical rule disjoint with $H(\pi)^+$, we arrive at the following results:

Lemma 7.10. *Let p be an atom. Then:*

- *$p \in B(\pi)^-$ if and only if for all $(I, J) \in \mathcal{X}$,*

$$p \in J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \ ;$$

- *$p \in H(\pi)^+$ if and only if $p \notin B(\pi)^-$ and for all $(I, J) \in \mathcal{X}$,*

$$p \in I \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \ .$$

Proof. See Appendix E, page 246. □

As can be derived from Lemma 7.9, the role of atoms from $H(\pi)^-$ and $B(\pi)^+$ is dual to that of atoms from $B(\pi)^-$ and $H(\pi)^+$. Intuitively, their *absence* in J, and sometimes also in I, implies that (I, J) is an SE-model of π. It follows from the first condition of Lemma E.23 that for every atom $p \in B(\pi)^+$ it holds that

$$p \notin I \wedge J \cap H(\pi)^+ \neq \emptyset \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \ . \tag{7.3}$$

Furthermore, the second condition in Lemma 7.9 implies that every $p \in B(\pi)^+ \cup H(\pi)^-$ satisfies the following condition:

$$p \notin J \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \ . \tag{7.4}$$

These observations lead to the following results:

Lemma 7.11. *Let p be an atom. Then:*

- *$p \in B(\pi)^+$ if and only if for all $(I, J) \in \mathfrak{X}$,*

$$(p \notin I \wedge J \cap H(\pi)^+ \neq \emptyset) \vee p \notin J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \ ;$$

- *$p \in H(\pi)^-$ if and only if $p \notin B(\pi)^+$ and for all $(I, J) \in \mathfrak{X}$,*

$$p \notin J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \ .$$

Proof. See Appendix E, page 247. □

Together, Lemmas 7.10 and 7.11 are sufficient to construct an SE-canonical rule from its set of SE-models. The following definition sums up these results by introducing the notion of *SE-induced rules*:

Definition 7.12 (Rule SE-Induced by a Set of Interpretations)**.** Let $\mathcal{M}$ be a set of three-valued interpretations. The *rule SE-induced by* $\mathcal{M}$, denoted by $\|\mathcal{M}\|_{\mathsf{SE}}$, is defined as follows: If $\mathcal{M} = \mathfrak{X}$, then $\|\mathcal{M}\|_{\mathsf{SE}} = \tau$; otherwise, $\|\mathcal{M}\|_{\mathsf{SE}}$ is of the form

$$H_{\mathsf{SE}}(\mathcal{M})^+; {\sim} H_{\mathsf{SE}}(\mathcal{M})^- \leftarrow B_{\mathsf{SE}}(\mathcal{M})^+, {\sim} B_{\mathsf{SE}}(\mathcal{M})^- .$$

where

$$\begin{aligned}
B_{\mathsf{SE}}(\mathcal{M})^- &= \{\, p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \in J \Longrightarrow (I, J) \in \mathcal{M} \,\} \ , \\
H_{\mathsf{SE}}(\mathcal{M})^+ &= \{\, p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \in I \Longrightarrow (I, J) \in \mathcal{M} \,\} \setminus B_{\mathsf{SE}}(\mathcal{M})^- \ , \\
B_{\mathsf{SE}}(\mathcal{M})^+ &= \left\{\, p \in \mathcal{A} \;\middle|\; \forall (I, J) \in \mathfrak{X} : (p \notin I \wedge J \cap H_{\mathsf{SE}}(\mathcal{M})^+ \neq \emptyset) \vee p \notin J \Longrightarrow (I, J) \in \mathcal{M} \,\right\} , \\
H_{\mathsf{SE}}(\mathcal{M})^- &= \{\, p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \notin J \Longrightarrow (I, J) \in \mathcal{M} \,\} \setminus B_{\mathsf{SE}}(\mathcal{M})^+ \ .
\end{aligned}$$

The main property of SE-induced rules is that every SE-canonical rule is induced by its own set of SE-models and can thus be "reconstructed" from it. This follows directly from Definition 7.12 using Lemmas 7.10 and 7.11.

Theorem 7.13. *For every SE-canonical rule π, $\|[\![\pi]\!]_{\mathsf{SE}}\|_{\mathsf{SE}} = \pi$.*

Proof. See Appendix E, page 247. □

This result, together with Theorem 7.8, has a number of consequences. First, for any rule π, the SE-canonical rule $\mathsf{can}_{\mathsf{SE}}(\pi)$ is SE-induced by the set of SE-models of π.

Corollary 7.14. *For every rule π, $\|[\![\pi]\!]_{\mathsf{SE}}\|_{\mathsf{SE}} = \mathsf{can}_{\mathsf{SE}}(\pi)$.*

Proof. Follows directly from Theorems 7.8 and 7.13. □

Furthermore, Theorem 7.13 implies that for two different SE-canonical rules π_1, π_2 we have $\|[\![\pi_1]\!]_{\mathsf{SE}}\|_{\mathsf{SE}} = \pi_1$ and $\|[\![\pi_2]\!]_{\mathsf{SE}}\|_{\mathsf{SE}} = \pi_2$, so $[\![\pi_1]\!]_{\mathsf{SE}}$ and $[\![\pi_2]\!]_{\mathsf{SE}}$ must differ.

Corollary 7.15. *No two different SE-canonical rules are SE-equivalent.*

Proof. Follows directly from Theorem 7.13. □

Finally, the previous corollary together with Theorem 7.8 imply that for every rule there not only exists an SE-equivalent SE-canonical rule, but this rule is also unique.

Corollary 7.16. *Every rule is SE-equivalent to exactly one SE-canonical rule.*

Proof. Follows directly from Theorem 7.8 and Corollary 7.15. □

7.1.3 SE-Rule-Expressible Sets of Interpretations

Naturally, not all sets of three-valued interpretations correspond to a single rule, otherwise any program could be reduced to a single rule. The conditions under which a set of three-valued interpretations is SE-rule-expressible are worth examining.

The set of SE-models of any program is well-defined, i.e. whenever it contains (I, J), it also contains (J, J). Also, every well-defined set of three-valued interpretations is the set of SE-models of some program (c.f. Proposition 6.7). We offer two analogical characterisations for the class of SE-rule-expressible sets of three-valued interpretations. The first is based on SE-induced rules defined in the previous section, while the second is formulated using lattice theory and is strongly related to Lemma 7.9.

The first characterisation follows from two properties of the $\| \cdot \|_{\mathsf{SE}}$ transformation. First, it can be applied to any set of three-valued interpretations, even those that are not SE-rule-expressible. Second, if $\|\mathcal{M}\|_{\mathsf{SE}} = \pi$, then it holds that $[\![\pi]\!]_{\mathsf{SE}}$ is a subset of $\mathcal{M}$.

Lemma 7.17. *The set of all SE-models of an SE-canonical rule π is the least among all sets of three-valued interpretations $\mathcal{M}$ such that $\|\mathcal{M}\|_{\mathsf{SE}} = \pi$.*

Proof. See Appendix E, page 247. □

Thus, to verify that $\mathcal{M}$ is SE-rule-expressible, it suffices to check that all interpretations from $\mathcal{M}$ are SE-models of the rule $\|\mathcal{M}\|_{\mathsf{SE}}$.

The second characterisation follows from Lemma 7.9 – if $\mathcal{M}$ is SE-rule-expressible, then its complement consists of three-valued interpretations (I, J) following a certain pattern. Their second component J always includes a fixed set of atoms and is itself included in another fixed set of atoms. Their first component I satisfies a similar property, but only if a certain further condition is satisfied by J. More formally, for the sets

$$I^{\perp} = B(\pi)^{+}, \quad I^{\top} = \mathcal{A} \setminus H(\pi)^{+}, \quad J^{\perp} = H(\pi)^{-} \cup B(\pi)^{+}, \quad J^{\top} = \mathcal{A} \setminus B(\pi)^{-},$$

it holds that all three-valued interpretations from the complement of $\mathcal{M}$ are of the form (I, J) where $J^{\perp} \subseteq J \subseteq J^{\top}$ and either $J \subseteq I^{\top}$ or $I^{\perp} \subseteq I \subseteq I^{\top}$. It turns out that this also holds vice versa: if the complement of $\mathcal{M}$ satisfies this property, then $\mathcal{M}$ is SE-rule-expressible. Furthermore, to accentuate the particular structure that arises, we can substitute the condition $J^{\perp} \subseteq J \subseteq J^{\top}$ with saying that J belongs to a convex sublattice of $\mathfrak{I}$.[1] A similar substitution can be performed for I, yielding:

[1] A sublattice L of L' is *convex* if $u \in L$ whenever $s, t \in L$ and $s \leq u \leq t$ holds in L'. For more details see e.g. (Davey and Priestley, 1990).

Theorem 7.18. *Let $\mathcal{M}$ be a set of three-valued interpretations. Then the following conditions are equivalent:*

1. *$\mathcal{M}$ is SE-rule-expressible.*
2. *$\mathcal{M} \subseteq [\![\|\mathcal{M}\|_{\mathsf{SE}}]\!]_{\mathsf{SE}}$.*
3. *There exist convex sublattices L_1, L_2 of $(\mathcal{I}, \subseteq)$ such that*

$$\mathcal{X} \setminus \mathcal{M} = \{ (I, J) \in \mathcal{X} \mid I \in L_1 \land J \in L_2 \} \cup \{ (I, J) \in \mathcal{X} \mid J \in L_1 \cap L_2 \} .^{2}$$

Proof. See Appendix E, page 249. □

7.2 Robust Equivalence Models

The results presented in Section 7.1 serve to facilitate the transitions back and forth between a rule and the set of its SE-models and make it possible to determine whether a given set of three-valued interpretations is the set of SE-models of some rule.

The introduction of SE-canonical rules, which form a set of representatives of rule equivalence classes induced by SE-models, also reveals the exact expressivity of SE-models semantics with respect to a single rule. On the one hand, SE-models strip away some irrelevant syntactic details, facilitating the manipulation of rules and programs.

On the other hand, a rule with a default literal in its head is indistinguishable from an integrity constraint under SE-models. For example, the rules

$$\leftarrow p, q. \qquad \sim p \leftarrow q. \qquad \sim q \leftarrow p. \tag{7.5}$$

have the same set of SE-models. In a static setting, these rules indeed carry essentially the same meaning: "it must not be the case that p and q are both true". But in a dynamic context, the latter two rules may, in addition, express that the truth of one atom gives a *reason* for the other atom to *cease being true*. For example, an update of the program $\{ p., q. \}$ by $\{ \sim p \leftarrow q. \}$ leads to the stable model $\{ q \}$ while an update by $\{ \sim q \leftarrow p. \}$ to the stable model $\{ p \}$. This convention is adopted by causal rejection-based rule update semantics, such as the AS-, JU-, DS- and RD-semantics, which constitute one of the most mature approaches to rule updates.

In order to be able to semantically characterise causal rejection-based rule update semantics, we need to distinguish between constraints and rules with default literals in their heads. These classes can be formally captured as follows:

Definition 7.19 (Constraint and Abolishing Rule)**.** A rule π is a *constraint* if $H(\pi) = \emptyset$ and $B(\pi)^+$ is disjoint with $B(\pi)^-$.[3]

A rule π is *abolishing* if $H(\pi)^+ = \emptyset$, $H(\pi)^- \neq \emptyset$ and the sets $H(\pi)^-$, $B(\pi)^+$ and $B(\pi)^-$ are pairwise disjoint.

So what we are looking for is a semantic characterisation of rules that

1) can distinguish constraints from related abolishing rules;
2) discards irrelevant syntactic details (akin to SE-models);
3) has a clear link to stable models (akin to SE-models).

In the following we introduce *robust equivalence models*, or *RE-models* for short, which exactly meet these criteria. They are defined as follows:

[2] Note that $\mathcal{X}$ is the set of all three-valued interpretations, as defined in Definition 6.1 on page 100.
[3] The latter condition guarantees that a constraint is not tautological.

Definition 7.20 (RE-Model)**.** Let π be a rule. A three-valued interpretation (I, J) is an *RE-model of* π if $I \models \pi^J$. The set of all RE-models of a rule π is denoted by $[\![\pi]\!]_{\mathsf{RE}}$ and for any program $\mathcal{P}$, $[\![\mathcal{P}]\!]_{\mathsf{RE}} = \bigcap_{\pi \in \mathcal{P}} [\![\pi]\!]_{\mathsf{RE}}$.

We say that π is *RE-tautological* if $[\![\pi]\!]_{\mathsf{RE}} = \mathfrak{X}$. Rules π, σ are *RE-equivalent* if $[\![\pi]\!]_{\mathsf{RE}} = [\![\sigma]\!]_{\mathsf{RE}}$.

Thus, unlike with SE-models, it is not required that $J \models \pi$ in order for (I, J) to be an RE-model of π. As a consequence, RE-models can distinguish between rules in (7.5): while both $(\{ q \}, \{ p, q \})$ and $(\{ p \}, \{ p, q \})$ are RE-models of the constraint, the former is not an RE-model of the first abolishing rule and the latter is not an RE-model of the second abolishing rule. This property holds in general, establishing requirement 1):

Proposition 7.21. *If π, σ are two different abolishing rules or an abolishing rule and a constraint, then π, σ are not RE-equivalent.*

Proof. See Appendix E, page 258. □

As for requirement 2), we first note that RE-equivalence is a refinement of SE-equivalence – there are no rules that are RE-equivalent but not SE-equivalent. The following result also shows that it is *only* the ability to distinguish between constraints and abolishing rules that is introduced by RE-models – rules that are not RE-equivalent to abolishing rules are distinguishable by RE-models if and only if they are distinguishable by SE-models. Furthermore, the classes of SE-tautological and RE-tautological rules coincide, so we can simply use the word *tautological* without ambiguity.

Proposition 7.22 (RE-Equivalence vs. SE-Equivalence)**.**

- *If two rules are RE-equivalent, then they are SE-equivalent.*
- *If two rules, neither of which is RE-equivalent to an abolishing rule, are SE-equivalent, then they are RE-equivalent.*
- *A rule is RE-tautological if and only if it is SE-tautological.*

Proof. See Appendix E, page 259. □

The affinity between SE-models and stable models is fully retained by RE-models, which establishes requirement 3).

Proposition 7.23 (RE-Models vs. Stable Models)**.** *An interpretation J is a stable model of a program $\mathcal{P}$ if and only if $(J, J) \in [\![\mathcal{P}]\!]_{\mathsf{RE}}$ and for all $I \subsetneq J$, $(I, J) \notin [\![\mathcal{P}]\!]_{\mathsf{RE}}$.*

Proof. See Appendix E, page 259. □

Also worth noting is that any set of three-valued interpretations can be expressed by a program using RE-models. This is not the case with SE-models since only well-defined sets of three-valued interpretations have corresponding programs.

Proposition 7.24. *Let $\mathcal{M}$ be a set of three-valued interpretations. Then there exists a program $\mathcal{P}$ such that $[\![\mathcal{P}]\!]_{\mathsf{RE}} = \mathcal{M}$.*

Proof. See Appendix E, page 259. □

Further properties of RE-models, analogous to those established in Section 7.1 for SE-models, can be found in Appendix E, starting on page 249. RE-models also seem to be closely related to *T-models* (Wong, 2007) used in the context of *forgetting* in logic programs.

In the following section we define new notions of program equivalence and entailment based on viewing a program as the set of sets of SE- or RE-models of its rules.

7.3 Program Equivalence and Entailment

While in Classical Logic equivalence under classical models is *the* equivalence, there is no such single notion of program equivalence. Despite that, suitable equivalence relations between logic programs are of great interest as they allow for the study of behaviour-preserving transformations and of expressiveness of different classes of programs – they open the doors towards deep semantic results about logic programs.

In the context of rule updates, a suitable notion of program equivalence would enable the definition of rule update operators that do not rely on rule syntax, yet they respect syntactic properties such as *support* and *fact update*.

In the following we review the relevant program equivalence relations that have been identified throughout the years (c.f. (Lifschitz et al., 2001; Inoue and Sakama, 2004; Woltran, 2008)) and define additional ones, based on the idea of viewing a logic program as the set of sets of SE- or RE-models of its rules. We also introduce the corresponding notions of program entailment and show a strength comparison of all equivalence and entailment relations considered.

7.3.1 Existing Notions of Program Equivalence

When considering logic programs under the stable models semantics, the first choice for an equivalence between two programs is *stable equivalence* (or *SM-equivalence*), i.e. two programs are equivalent when they have the same stable models. In many cases, however, SM-equivalence is not strong enough because programs with the same stable models, when augmented with the same additional rules, may end up having completely different stable models.

This motivates the use of *strong equivalence* (Lifschitz et al., 2001) which requires that stable models stay the same even in the presence of additional rules. As mentioned in Section 6.1, programs are strongly equivalent if and only if they have the same set of SE-models. Hence we also refer to strong equivalence as *SE-equivalence*.

However, as shown in Chapter 6, SE-equivalence is not satisfactory when used as a basis for rule updates: operators that satisfy the syntax-independence postulate $(\text{P4.1})_{\text{SE}}$ are out of line with elementary intuitions regarding rule updates (c.f. Theorem 6.18).

So, in order to arrive at plausible semantic rule update operators, we need to search for a notion of program equivalence that is stronger than SE-equivalence. One candidate for this is *strong update equivalence* (or *SU-equivalence*) (Inoue and Sakama, 2004), which requires that under both additions and removals of rules, stable models of the two programs in question remain the same. It has been shown in (Inoue and Sakama, 2004) that this notion of equivalence is very strong – programs are SU-equivalent only if they contain exactly the same non-tautological rules and, in addition, each of them may contain some tautological ones. Formally, programs $\mathcal{P}$, $\mathcal{Q}$ are SU-equivalent if and only if $[\![(\mathcal{P} \setminus \mathcal{Q}) \cup (\mathcal{Q} \setminus \mathcal{P})]\!]_{\text{SE}} = \mathcal{X}$. Thus SU-equivalence seems perhaps *too strong* as it is not difficult to find rules such as $(\sim p \leftarrow p.)$ and $(\leftarrow p.)$ that are syntactically different but seem to carry the same meaning, even when subject to updates. In order to weaken SU-equivalence, we can use SE- or RE-models to filter out irrelevant syntactic details in such rules.

7.3.2 Rules as Sets of SE- and RE-Models

This line of thought brings us to the possibility of viewing a program $\mathcal{P}$ as the set of sets of models of its rules. For this purpose we introduce the following notation:

Definition 7.25 (Set of Sets of Models of a Program)**.** Let $\mathcal{P}$ be a program. Then

$$\langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{SE}} = \{\, [\![\pi]\!]_{\mathsf{SE}} \mid \pi \in \mathcal{P} \,\} \qquad \text{and} \qquad \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}} = \{\, [\![\pi]\!]_{\mathsf{RE}} \mid \pi \in \mathcal{P} \,\} \enspace .$$

We can now say that programs $\mathcal{P}$, $\mathcal{Q}$ are equivalent when $\langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{SE}} = \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{SE}}$ or $\langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}$. This amounts to saying that two programs are equivalent when they contain the same rules modulo SE- and RE-models, respectively. Intuitively, the new notions of equivalence should be stronger than SE-equivalence but weaker than SU-equivalence.

Nevertheless, such a definition would not be as intended because the programs $\mathcal{P} = \emptyset$ and $\mathcal{Q} = \{\, \tau \,\}$ would not be considered equivalent even though they bear the same meaning and are both SE- and SU-equivalent. The solution is to add the canonical tautology to both $\mathcal{P}$ and $\mathcal{Q}$ before comparing their sets of sets of models. Thus we arrive at the following conditions for *strong rule equivalence* (*SR-equivalence* for short) and *robust rule equivalence* (*RR-equivalence* for short):

$$\langle\!\langle \mathcal{P} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{SE}} = \langle\!\langle \mathcal{Q} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{SE}} \enspace , \qquad \langle\!\langle \mathcal{P} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{Q} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

An important question is whether these notions of equivalence could still be too strong. As it turns out, in the context of updates, programs such as $\mathcal{P}_0 = \{\, p. \,\}$ and $\mathcal{Q}_0 = \{\, p., p \leftarrow q. \,\}$ are frequently treated the same way because the extra rule in $\mathcal{Q}_0$ is just a weakened version of the rule in $\mathcal{P}_0$. For instance, the notion of *update equivalence* introduced in (Leite, 2003), which is based on a particular approach to logic program updates, considers programs $\mathcal{P}_0$ and $\mathcal{Q}_0$ as equivalent because the extra rule in $\mathcal{Q}_0$ cannot influence the result of any subsequent updates. Since these programs are not SR- nor RR-equivalent, we also introduce the notions of *strong minimal rule equivalence* (or *SMR-equivalence*) and *robust minimal rule equivalence* (or *RMR-equivalence*) by considering only the subset-minimal sets of models from the two programs. Formally, the conditions for SMR- and RMR-equivalence between programs $\mathcal{P}$, $\mathcal{Q}$ are:

$$\min \langle\!\langle \mathcal{P} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{SE}} = \min \langle\!\langle \mathcal{Q} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{SE}} \enspace , \qquad \min \langle\!\langle \mathcal{P} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{RE}} = \min \langle\!\langle \mathcal{Q} \cup \{\, \tau \,\} \rangle\!\rangle_{\mathsf{RE}} \enspace ,$$

where $\min \mathcal{S}$ denotes the subset-minimal elements of a set of sets $\mathcal{S}$. Note that the programs $\mathcal{P}_0$, $\mathcal{Q}_0$ discussed above are both SMR- and RMR-equivalent.

The following definition formally establishes all of the mentioned program equivalences and, for completeness, defines *RE-equivalence* similarly to SE-equivalence.

Definition 7.26 (Program Equivalence)**.** Let $\mathcal{P}, \mathcal{Q}$ be programs, $\mathcal{P}^\tau = \mathcal{P} \cup \{\, \tau \,\}$, $\mathcal{Q}^\tau = \mathcal{Q} \cup \{\, \tau \,\}$ and let $\min \mathcal{S}$ denote the subset-minimal elements of any set of sets $\mathcal{S}$. We write

$\mathcal{P} \equiv_{\mathsf{SM}} \mathcal{Q}$	whenever	$[\![\mathcal{P}]\!]_{\mathsf{SM}} = [\![\mathcal{Q}]\!]_{\mathsf{SM}}$;
$\mathcal{P} \equiv_{\mathsf{SE}} \mathcal{Q}$	whenever	$[\![\mathcal{P}]\!]_{\mathsf{SE}} = [\![\mathcal{Q}]\!]_{\mathsf{SE}}$;
$\mathcal{P} \equiv_{\mathsf{RE}} \mathcal{Q}$	whenever	$[\![\mathcal{P}]\!]_{\mathsf{RE}} = [\![\mathcal{Q}]\!]_{\mathsf{RE}}$;
$\mathcal{P} \equiv_{\mathsf{SMR}} \mathcal{Q}$	whenever	$\min \langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{SE}} = \min \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{SE}}$;
$\mathcal{P} \equiv_{\mathsf{RMR}} \mathcal{Q}$	whenever	$\min \langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{RE}} = \min \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{RE}}$;
$\mathcal{P} \equiv_{\mathsf{SR}} \mathcal{Q}$	whenever	$\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{SE}} = \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{SE}}$;
$\mathcal{P} \equiv_{\mathsf{RR}} \mathcal{Q}$	whenever	$\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{RE}}$.
$\mathcal{P} \equiv_{\mathsf{SU}} \mathcal{Q}$	whenever	$[\![(\mathcal{P} \setminus \mathcal{Q}) \cup (\mathcal{Q} \setminus \mathcal{P})]\!]_{\mathsf{SE}} = \mathfrak{X}$.

We say that $\mathcal{P}$ *is X-equivalent to* $\mathcal{Q}$ if $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$.

7.3.3 Program Entailment

In order to consider belief update principles in the context of rule updates, we also need to establish notions of *program entailment* which are in line with the above defined program equivalence relations. This task is troublesome in case of SM-equivalence because the usage of entailment in belief update postulates is clearly a monotonic one while stable models are non-monotonic. For instance, a reformulation of (B1) would require that $\mathcal{P} \oplus \mathcal{U} \models \mathcal{U}$, though there is no reason for $\mathcal{P} \oplus \mathcal{U}$ to have *less* stable models than (or the same as) $\mathcal{U}$. Due to these issues, we refrain from defining SM-entailment. The remaining entailment relations are defined as follows:

Definition 7.27 (Program Entailment)**.** Let $\mathcal{P}, \mathcal{Q}$ be programs, $\mathcal{P}^\tau = \mathcal{P} \cup \{ \tau \}$ and $\mathcal{Q}^\tau = \mathcal{Q} \cup \{ \tau \}$. We write

$\mathcal{P} \models_{\mathsf{SE}} \mathcal{Q}$	whenever	$[\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{SE}}$;
$\mathcal{P} \models_{\mathsf{RE}} \mathcal{Q}$	whenever	$[\![\mathcal{P}]\!]_{\mathsf{RE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{RE}}$;
$\mathcal{P} \models_{\mathsf{SMR}} \mathcal{Q}$	whenever	$\forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}}$;
$\mathcal{P} \models_{\mathsf{RMR}} \mathcal{Q}$	whenever	$\forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}}$;
$\mathcal{P} \models_{\mathsf{SR}} \mathcal{Q}$	whenever	$\forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{SE}} = [\![\sigma]\!]_{\mathsf{SE}}$;
$\mathcal{P} \models_{\mathsf{RR}} \mathcal{Q}$	whenever	$\forall \sigma \in \mathcal{Q}^\tau \, \exists \pi \in \mathcal{P}^\tau : [\![\pi]\!]_{\mathsf{RE}} = [\![\sigma]\!]_{\mathsf{RE}}$;
$\mathcal{P} \models_{\mathsf{SU}} \mathcal{Q}$	whenever	$[\![\mathcal{Q} \setminus \mathcal{P}]\!]_{\mathsf{SE}} = \mathfrak{X}$.

We say that $\mathcal{P}$ *X-entails* $\mathcal{Q}$ if $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$.

As the following proposition shows, the defined entailment relations are fully in line with the respective equivalence relations.

Proposition 7.28. *Let X be one of SE, RE, SMR, RMR, SR, RR, SU and $\mathcal{P}$, $\mathcal{Q}$ be programs. Then,*

$$\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q} \qquad \textit{if and only if} \qquad \mathcal{P} \models_{\mathsf{X}} \mathcal{Q} \textit{ and } \mathcal{Q} \models_{\mathsf{X}} \mathcal{P}.$$

Proof. See Appendix E, page 260. □

7.3.4 Strength Comparison

Our previous considerations show that SM-equivalence is the weakest notion of program equivalence, followed by SE- and RE-equivalence, then by SMR- and RMR-equivalence and by SR- and RR-equivalence and finally SU-equivalence is the strongest program equivalence relation. To formally capture these relationships, we introduce the following concepts:

Definition 7.29 (Strength of Program Equivalence and Entailment)**.** Let $\equiv_{\mathsf{X}}, \equiv_{\mathsf{Y}}$ be equivalence relations and $\models_{\mathsf{X}}, \models_{\mathsf{Y}}$ preorders on the set of all programs. We write

- $\equiv_{\mathsf{X}} \preceq \equiv_{\mathsf{Y}}$ if $\mathcal{P} \equiv_{\mathsf{Y}} \mathcal{Q}$ implies $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$ for all programs $\mathcal{P}, \mathcal{Q}$;
- $\equiv_{\mathsf{X}} \prec \equiv_{\mathsf{Y}}$ if $\equiv_{\mathsf{X}} \preceq \equiv_{\mathsf{Y}}$ but not $\equiv_{\mathsf{Y}} \preceq \equiv_{\mathsf{X}}$;
- $\models_{\mathsf{X}} \preceq \models_{\mathsf{Y}}$ if $\mathcal{P} \models_{\mathsf{Y}} \mathcal{Q}$ implies $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ for all programs $\mathcal{P}, \mathcal{Q}$;
- $\models_{\mathsf{X}} \prec \models_{\mathsf{Y}}$ if $\models_{\mathsf{X}} \preceq \models_{\mathsf{Y}}$ but not $\models_{\mathsf{Y}} \preceq \models_{\mathsf{X}}$.

The strength comparison of the defined notions of equivalence and entailment is depicted in Figure 7.1 and formally stated in the following proposition:

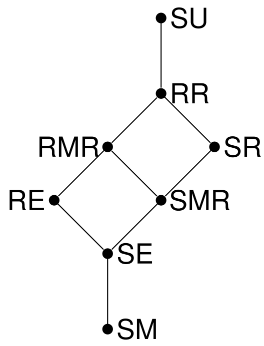

Figure 7.1: Notions of program equivalence and entailment from the weakest in the bottom to the strongest on top. A missing link between X and Y indicates that $\equiv_{\mathsf{X}}$ is incomparable with $\equiv_{\mathsf{Y}}$ and $\models_{\mathsf{X}}$ is incomparable with $\models_{\mathsf{Y}}$.

Proposition 7.30. *The following holds:*

(1) $\equiv_{\mathsf{SM}} \prec \equiv_{\mathsf{SE}} \prec \equiv_{\mathsf{RE}} \prec \equiv_{\mathsf{RMR}} \prec \equiv_{\mathsf{RR}} \prec \equiv_{\mathsf{SU}}$ *and* $\models_{\mathsf{SE}} \prec \models_{\mathsf{RE}} \prec \models_{\mathsf{RMR}} \prec \models_{\mathsf{RR}} \prec \models_{\mathsf{SU}}$ *;*

(2) $\equiv_{\mathsf{SE}} \prec \equiv_{\mathsf{SMR}} \prec \equiv_{\mathsf{SR}} \prec \equiv_{\mathsf{RR}}$ *and* $\models_{\mathsf{SE}} \prec \models_{\mathsf{SMR}} \prec \models_{\mathsf{SR}} \prec \models_{\mathsf{RR}}$ *;*

(3) $\equiv_{\mathsf{SMR}} \prec \equiv_{\mathsf{RMR}}$ *and* $\models_{\mathsf{SMR}} \prec \models_{\mathsf{RMR}}$ *;*

(4) $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SMR}}$ *and* $\equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{RE}}$ *and* $\models_{\mathsf{RE}} \not\preceq \models_{\mathsf{SMR}}$ *and* $\models_{\mathsf{SMR}} \not\preceq \models_{\mathsf{RE}}$ *;*

(5) $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SR}}$ *and* $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RE}}$ *and* $\models_{\mathsf{RE}} \not\preceq \models_{\mathsf{SR}}$ *and* $\models_{\mathsf{SR}} \not\preceq \models_{\mathsf{RE}}$ *;*

(6) $\equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{SR}}$ *and* $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RMR}}$ *and* $\models_{\mathsf{RMR}} \not\preceq \models_{\mathsf{SR}}$ *and* $\models_{\mathsf{SR}} \not\preceq \models_{\mathsf{RMR}}$ *.*

Proof. See Appendix E, page 262. □

7.4 Discussion

In this chapter we have continued our search for a suitable semantic approach to rule updates. Due to the conclusions of Chapter 6 which revealed the limitations of SE-models when used as a basis for rule updates, here we focused on richer semantic characterisations of logic programs.

First we pinpointed the expressivity of SE-models w.r.t. single rules and proposed RE-models as an alternative monotonic characterisation of rules that is capable of distinguishing constraints from abolishing rules. Then we defined a number of program equivalence and entailment relations based on the idea of viewing a program as the set of sets of models of its rules, resulting in the notions of SR- and RR-equivalence as well as their weaker versions, SMR- and RMR-equivalence. We also introduced corresponding program entailment relations.

In terms of strength, the new notions of equivalence fall between SE-equivalence and the very strong SU-equivalence. Thus they seem to be interesting candidates for a semantic basis of rule updates, especially RR- and RMR-equivalence which, due to the properties of RE-models, seem sufficiently expressive to capture causal rejection-based rule update semantics.

The following chapter confirms this conjecture by describing a general semantic approach to rule updates based on RR-equivalence, and showing that its instances directly correspond with the JU-semantics for rule updates.

8

Exception-Based Updates

JOÃO: In the time you have left of your PhD, you should follow this trail and see where it leads.

MARTIN: But what if I don't get anywhere?

JOÃO: Then you report on the obstacles you encountered.

JUNE 2011, LISBON, PORTUGAL

Up until now we have seen, in Chapter 6, that even though SE-models can be used as a basis for semantic rule update operators, they cannot be combined with fundamental properties of traditional approaches to rule updates: *support* and *fact update*.

This led us to the study of richer semantic characterisations of rules and programs in Chapter 7. We defined *RE-models*, a monotonic semantics that retains many important properties of SE-models and is able to differentiate additional classes of rules. Since these classes are handled differently in causal rejection-based approaches to rule updates, the use of RE-models instead of SE-models is necessary if we want to characterise them semantically.

Subsequently, we introduced the idea of viewing a program as the *set of sets of models of its rules*, acknowledging rules as the atomic pieces of knowledge in a program and, at the same time, abstracting away from irrelevant differences between their syntactic forms, focusing on their semantic content. We defined notions of program equivalence and entailment that follow from this approach. In particular, when using RE-models as the semantics for individual rules, this naturally leads to the concept of RR-equivalence and its weaker version, RMR-equivalence.

In this chapter we propose a generic method for defining semantic update operators: a program, viewed as the *set of sets of RE-models of its rules*, is updated by introducing additional interpretations – *exceptions* – to the original sets of RE-models. Using this approach we are able to capture a traditional approach to rule updates, combining *syntax-independence* with a range of desirable properties. Moreover, the same ideas can be used to capture a range of *belief update operators*. We thus bridge the two different update paradigms in a single framework.

More specifically, our main contributions are as follows:

- we show that particular exception-based rule update operators satisfy fundamental syntactic properties of rule updates such as *support*, *fact update* and *causal rejection*;
- we define exception-based operators that offer a direct semantic counterpart of the *JU-semantics* for rule updates and shed new light on the problem of *state condensing*;
- we provide an exhaustive analysis of *semantic properties* of exception-based rule updates, showing that they naturally satisfy a range of desirable properties such as *immunity to tautological updates*, *syntax-independence*, and counterparts of many *belief update postulates* w.r.t. RR-equivalence;
- we generalise these ideas and show that exception-based operators can capture many *model-based* and *formula-based* belief update operators.

The remainder of this chapter is structured as follows: In Section 8.1 we introduce exception-based rule update operators and examine their syntactic as well as semantic properties. In Section 8.2 we define an abstract framework for exception-based operators and show that it can capture a range of belief update operators. We discuss our findings and future research directions in Section 8.4.

The relevant proofs are provided in Appendix F. Parts of this chapter have been published in (Slota and Leite, 2012a,b).

8.1 Exception-Based Rule Update Operators

In this section we propose a generic framework for defining semantic rule update operators. We define instances of the framework and show that they enjoy a number of plausible properties, ranging from the respect for support and fact update to syntax-independence and other semantic properties.

As suggested above, a program is semantically characterised by the set of sets of RE-models of its rules. Our update framework is based on a simple yet novel idea of introducing additional interpretations – *exceptions* – to the sets of RE-models of rules in the original program. The formalisation of this idea is straight-forward: an exception-based update operator is characterised by an *exception function* ε that takes three inputs: the set of RE-models $[\![\pi]\!]_{\mathsf{RE}}$ of a rule $\pi \in \mathcal{P}$ and the semantic characterisations $\langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}$, $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ of the original and updating programs. It then returns the three-valued interpretations that are to be introduced as exceptions to π, so the characterisation of the updated program contains the augmented set of RE-models

$$[\![\pi]\!]_{\mathsf{RE}} \cup \varepsilon\left([\![\pi]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}\right) \quad . \tag{8.1}$$

Hence the semantic characterisation of $\mathcal{P}$ updated by $\mathcal{U}$ is

$$\{\, [\![\pi]\!]_{\mathsf{RE}} \cup \varepsilon\left([\![\pi]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}\right) \mid \pi \in \mathcal{P} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \quad . \tag{8.2}$$

In other words, the set of RE-models of each rule π from $\mathcal{P}$ is augmented with the respective exceptions while the sets of RE-models of rules from $\mathcal{U}$ are kept untouched.

From the syntactic viewpoint, we want a rule update operator $\oplus$ to return a program

$\mathcal{P} \oplus \mathcal{U}$ with the semantic characterisation (8.2). This brings us to the following issue: What if no rule exists whose set of RE-models is equal to (8.1)? In that case, no rule corresponds to the augmented set of RE-models of a rule $\pi \in \mathcal{P}$, so the program $\mathcal{P} \oplus \mathcal{U}$ cannot be constructed. Moreover, such situations may occur quite frequently since a single rule has very limited expressivity. For instance, updating the fact $(p.)$ by the rule $(\sim p \leftarrow q, r.)$ may easily result in a set of RE-models expressible by the program $\{ p \leftarrow \sim q., p \leftarrow \sim r. \}$ but not expressible by any single rule. To keep a firm link to operations on syntactic objects, we henceforth deal with this problem by allowing the inputs and output of rule update operators to be sets of rules *and programs*, which we dub *rule bases*.[1] In other words, the result of updating a rule, i.e. introducing exceptions to it, may be a set of rules, so the result of updating a program may be a rule base. Technically, a rule base can capture any possible result of an exception-based update due to Proposition 7.24).

Definition 8.1 (Rule Base)**.** A *rule base* is any set of rules and programs. For every rule base $\mathcal{R}$ we define the following:

- an interpretation J is a *model of* $\mathcal{R}$, denoted by $J \models \mathcal{R}$, if $J \models \Pi$ for all $\Pi \in \mathcal{R}$;
- $\mathcal{R}^J = \{ \Pi^J \mid \Pi \in \mathcal{R} \}$ for every interpretation J;
- an interpretation J is a *stable model of* $\mathcal{R}$ if J is a subset-minimal model of $\mathcal{R}^J$;
- the set of stable models of $\mathcal{R}$ is denoted by $[\![\mathcal{R}]\!]_{\mathsf{SM}}$;
- $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{SE}} = \{ [\![\Pi]\!]_{\mathsf{SE}} \mid \Pi \in \mathcal{R} \}$ and $[\![\mathcal{R}]\!]_{\mathsf{SE}} = \bigcap \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{SE}}$;
- $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}} = \{ [\![\Pi]\!]_{\mathsf{RE}} \mid \Pi \in \mathcal{R} \}$ and $[\![\mathcal{R}]\!]_{\mathsf{RE}} = \bigcap \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$.

All notions of program equivalence and entailment are extended to rule bases by using the same definition.

Note that a program is a special case of a rule base. Each element Π of a rule base, be it a rule or a program, represents an *atomic* piece of information. Exception-based update operators view and manipulate Π only through its set of RE-models $[\![\Pi]\!]_{\mathsf{RE}}$. Due to this, we refer to all such elements Π as *rules*, even if formally they may actually be programs.

Having resolved this issue, we can proceed to the definition of an exception-based rule update operator.

Definition 8.2 (Exception-Based Rule Update Operator)**.** A rule update operator $\oplus$ is *exception-based* if for some exception function ε, $\langle\!\langle \mathcal{R} \oplus \mathcal{U} \rangle\!\rangle$ is equal to (8.2) for all rule bases $\mathcal{R}, \mathcal{U}$. In that case we also say that $\oplus$ is *ε-based*.

Note that for each exception function ε there is a whole class of ε-based rule update operators that differ in the syntactic representations of the sets of RE-models in (8.2). For instance, when working over the set of propositional symbols $\mathcal{A} = \{ p, q \}$ and considering some ε-based operator $\oplus$, the exception function may specify that for some programs $\mathcal{P}, \mathcal{U}$, the program $\mathcal{P} \oplus \mathcal{U}$ contains some rule or program representing the set of RE-models

$$\mathcal{M} = \{ (\emptyset, \emptyset), (\emptyset, p), (p, p), (\emptyset, q), (\emptyset, pq), (p, pq), (pq, pq) \} \ .[2]$$

This set can be represented by the rule $\pi = (p \leftarrow q.)$ or, alternatively, by the rule $\sigma = (p; \sim q \leftarrow q.)$, and the exception function does not specify which syntactic representation of the set should be used in $\mathcal{P} \oplus \mathcal{U}$.

[1] We allow for individual rules in a rule base out of convenience only. A single rule π in a rule base $\mathcal{R}$ is treated exactly the same way as if $\mathcal{R}$ contained the singleton program $\{ \pi \}$.

[2] We sometimes omit the usual set notation when we write interpretations. For example, instead of $\{ p, q \}$ we may simply write pq.

8.1.1 Simple Exception Functions and Syntactic Properties

Of particular interest to us is a constrained class of exception functions that requires less information to determine the resulting exceptions. Not only does it lead to simpler definitions and to modular, more efficient implementations, but the study of restricted classes of exception functions is also essential in order to understand their expressivity, i.e. the types of update operators they are able to capture. We focus on exception functions that produce exceptions based on conflicts between pairs of rules, one from the original and one from the updating program, while ignoring the context in which these rules are situated. More formally:

Definition 8.3 (Simple Exception Function)**.** An exception function ε is *simple* if for all $\mathcal{M} \subseteq \mathcal{X}$ and $\mathcal{S}, \mathcal{T} \subseteq 2^{\mathcal{X}}$,

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = \bigcup_{\mathcal{N} \in \mathcal{T}} \delta(\mathcal{M}, \mathcal{N})$$

where $\delta : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ is a *local exception function*. If $\oplus$ is an ε-based rule update operator, then we also say that $\oplus$ is *δ-based*, that *δ generates* $\oplus$ and that $\oplus$ is *simple*.

As we shall see, in spite of their local nature, particular simple exception functions generate rule update operators that satisfy the syntactic properties of rule update semantics that have been defined in Section 2.8.4 and are closely related to the JU-semantics for DLPs.

The inspiration for defining concrete local exception functions δ comes from rule update semantics based on causal rejection. But since the relevant concepts, such as that of a *conflict* or *rule rejection*, rely on rule syntax to which an exception function has no direct access, our first objective is to find similar concepts on the semantic level. In particular, we need to define conditions under which two sets of RE-models are in conflict. We define these conflicts w.r.t. a two-valued interpretation. We first introduce two preparatory concepts.

We define a *truth value substitution* as follows: Given an interpretation J, an atom p and a truth value $\mathsf{V} \in \{ \mathsf{T}, \mathsf{U}, \mathsf{F} \}$, by $J[\mathsf{V}/p]$ we denote the three-valued interpretation X such that $X(p) = \mathsf{V}$ and $X(q) = J(q)$ for all atoms $q \neq p$.

This enables us to introduce the main concept needed for defining a conflict between two sets of three-valued interpretations. Given a set of three-valued interpretations $\mathcal{M}$, an atom p, a truth value V_0 and a two-valued interpretation J, we say that $\mathcal{M}$ *forces p to have the truth value* V_0 *w.r.t.* J, denoted by $\mathcal{M}^J(p) = \mathsf{V}_0$, if

$$J[\mathsf{V}/p] \in \mathcal{M} \text{ if and only if } \mathsf{V} = \mathsf{V}_0 \ .$$

In other words, the three-valued interpretation $J[\mathsf{V}_0/p]$ must be the unique member of $\mathcal{M}$ that either coincides with J or differs from it only in the truth value of p. Note that $\mathcal{M}^J(p)$ stays undefined in case no V_0 with the above property exists.

Two sets of three-valued interpretations $\mathcal{M}, \mathcal{N}$ are *in conflict on atom p w.r.t. J*, denoted by $\mathcal{M} \bowtie_p^J \mathcal{N}$, if both $\mathcal{M}^J(p)$ and $\mathcal{N}^J(p)$ are defined and $\mathcal{M}^J(p) \neq \mathcal{N}^J(p)$. The following example illustrates all these concepts.

Example 8.4. Consider rules $\pi_0 = (p.)$, $\pi_1 = (\sim p \leftarrow \sim q.)$ with the respective sets of RE-models

$$\mathcal{M}_0 = \{ (p,p), (p,pq), (pq,pq) \} \ ,$$
$$\mathcal{M}_1 = \{ (\emptyset,\emptyset), (\emptyset,q), (q,q), (\emptyset,pq), (p,pq), (q,pq), (pq,pq) \} \ .$$

Intuitively, $\mathcal{M}_0$ forces p to T w.r.t. all interpretations and π_1 forces p to F w.r.t. interpretations in which q is false. Formally it follows that $\mathcal{M}_0^{\emptyset}(p) = \mathsf{T}$ because (p, p) belongs to $\mathcal{M}_0$ and neither $(\emptyset, p)$ nor $(\emptyset, \emptyset)$ belongs to $\mathcal{M}_0$. Similarly, it follows that $\mathcal{M}_1^{\emptyset}(p) = \mathsf{F}$. Hence $\mathcal{M}_0 \bowtie_p^{\emptyset} \mathcal{M}_1$. Using similar arguments we can conclude that $\mathcal{M}_0 \bowtie_p^{p} \mathcal{M}_1$. However, it does not hold that $\mathcal{M}_0 \bowtie_p^{pq} \mathcal{M}_1$ because $\mathcal{M}_1^{pq}(p)$ is undefined.

We are now ready to introduce the local exception function δ_{a}.

Definition 8.5 (Local Exception Function δ_{a})**.** The local exception function δ_{a} is for all $\mathcal{M}, \mathcal{N} \subseteq \mathfrak{X}$ defined as

$$\delta_{\mathsf{a}}(\mathcal{M}, \mathcal{N}) = \left\{ (I, J) \in \mathfrak{X} \;\middle|\; \exists p : \mathcal{M} \bowtie_p^J \mathcal{N} \right\} \quad .$$

Thus if there is a conflict on some atom w.r.t. J, the exceptions introduced by δ_{a} are of the form (I, J) where I can be an arbitrary subset of J. This means that δ_{a} introduces as exceptions all three-valued interpretations that preserve false atoms from J while the atoms that are true in J may be either true or undefined. This is somewhat related to the definition of a stable model where the default assumptions (false atoms) are fixed while the necessary truth of the remaining atoms is checked against the rules of the program. The syntactic properties of δ_{a}-based operators are as follows.

Theorem 8.6 (Syntactic Properties of δ_{a})**.** *Every δ_{a}-based rule update operator respects support and fact update. Furthermore, it also respects causal rejection and acyclic justified update w.r.t. DLPs of length at most two.*

Proof. See Appendix F.1, page 275. □

This means that δ_{a}-based rule update operators enjoy a combination of desirable syntactic properties that operators based on SE-models cannot (c.f. Theorem 6.18). However, these operators diverge from causal rejection, even on acyclic DLPs, when more than one update is performed.

Example 8.7. Consider again the rules π_0, π_1 and their sets of RE-models $\mathcal{M}_0$, $\mathcal{M}_1$ from Example 8.4 and some δ_{a}-based rule update operator $\oplus$. Then $\langle\!\langle \{ \pi_0 \} \oplus \{ \pi_1 \} \rangle\!\rangle_{\mathsf{RE}}$ will contain two elements: $\mathcal{M}_0'$ and $\mathcal{M}_1$, where

$$\mathcal{M}_0' = \mathcal{M}_0 \cup \delta_{\mathsf{a}}(\mathcal{M}_0, \mathcal{M}_1) = \mathcal{M}_0 \cup \{ (\emptyset, \emptyset), (\emptyset, p) \} \quad .$$

An additional update by the fact $\{ q. \}$ then leads to the characterisation

$$\left\langle\!\!\left\langle \bigoplus \langle \{ \pi_0 \}, \{ \pi_1 \}, \{ q. \} \rangle \right\rangle\!\!\right\rangle_{\mathsf{RE}}$$

which contains three elements: $\mathcal{M}_0''$, $\mathcal{M}_1$ and $\mathcal{M}_2$ where

$$\mathcal{M}_0'' = \mathcal{M}_0' \cup \{ (\emptyset, q), (q, q) \}$$

and $\mathcal{M}_2$ is the set of RE-models of $(q.)$.

Furthermore, due to Proposition 7.23, the interpretation $J = \{ q \}$ is a stable model of the program $\bigoplus \langle \{ \pi_0 \}, \{ \pi_1 \}, \{ q. \} \rangle$ because (q, q) belongs to all sets of models in the set of sets of models $\langle\!\langle \bigoplus \langle \{ \pi_0 \}, \{ \pi_1 \}, \{ q. \} \rangle \rangle\!\rangle_{\mathsf{RE}}$ and $(\emptyset, q)$ does not belong to $\mathcal{M}_2$. However, J does not respect causal rejection and it is not a JU-model of $(\{ \pi_0 \}, \{ \pi_1 \}, \{ q. \})$.

This shortcoming of δ_{a} can be overcome as follows:

Definition 8.8 (Local Exception Functions δ_b, δ_c)**.** The local exception functions δ_b, δ_c are for all $\mathcal{M}, \mathcal{N} \subseteq \mathfrak{X}$ defined as

$$\delta_b(\mathcal{M}, \mathcal{N}) = \{ (I, K) \in \mathfrak{X} \mid \exists J \, \exists p : \mathcal{M} \bowtie_p^J \mathcal{N} \wedge I \subseteq J \subseteq K \wedge (p \in K \setminus I \Longrightarrow K = J) \} \quad ,$$

$$\delta_c(\mathcal{M}, \mathcal{N}) = \begin{cases} \mathfrak{X} & \text{if } \mathcal{M} = \mathcal{N} \ ; \\ \delta_b(\mathcal{M}, \mathcal{N}) & \text{otherwise} \ . \end{cases}$$

The functions δ_b and δ_c introduce more exceptions than δ_a. A conflict on p w.r.t. J leads to the introduction of interpretations in which atoms either maintain the truth value they had in J, or they become undefined. They must also satisfy an extra condition: when p becomes undefined, no other atom may pass from false to undefined. Interestingly, this leads to operators that satisfy all syntactic properties.

Theorem 8.9 (Syntactic Properties of δ_b and δ_c)**.** *Let $\oplus$ be a δ_b- or δ_c-based rule update operator. Then $\oplus$ respects support, fact update, causal rejection and acyclic justified update.*

Proof. See Appendix F.1, page 281. □

The difference between δ_b and δ_c is in that δ_c additionally "wipes out" rules from the original program that are repeated in the update by introducing all interpretations as exceptions to them, rendering them tautological. This will become more pronounced later when we examine semantic properties of simple exception-based rule update operators.

Before that, however, it is worth noting that δ_b- and δ_c-based operators are very closely related to the JU-semantics, even on programs with cycles. They diverge from it only on rules with an appearance of the same atom in both the head and body. Formally, we say that a rule is a *local cycle* if $(H(\pi)^+ \cup H(\pi)^-) \cap (B(\pi)^+ \cup B(\pi)^-) \neq \emptyset$.

Theorem 8.10. *Let $\boldsymbol{P}$ be a DLP, J an interpretation and $\oplus$ a δ_b- or δ_c-based rule update operator. Then,*

- $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{SM}} \subseteq [\![\boldsymbol{P}]\!]_{\mathsf{JU}}$ *and*
- *if* $\mathsf{all}(\boldsymbol{P})$ *contains no local cycles, then* $[\![\boldsymbol{P}]\!]_{\mathsf{JU}} \subseteq [\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{SM}}$.

Proof. See Appendix F.1, page 281. □

This means that up to the marginal case of local cycles, δ_b and δ_c can be seen as semantic characterisations of the JU-semantics: they lead to stable models that, typically, coincide with JU-models. This tight relationship also sheds new light on the problem of *state condensing* where the goal is to transform a DLP into a single program over the same alphabet that would behave just as the original DLP when further updates are performed. While this cannot be done if the result must be a non-disjunctive program (Leite, 2003), it follows from Theorem 8.10 that a rule base is sufficiently expressive. It stays an open question whether a disjunctive program would suffice instead of a rule base or not.

Corollary 8.11 (State Condensing into a Rule Base)**.** *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP such that $\mathsf{all}(\boldsymbol{P})$ contains no local cycles, $\oplus$ be a δ_b- or δ_c-based rule update operator and $j < n$. Then there exists a rule base $\mathcal{R}$ such that $[\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\bigoplus \boldsymbol{P}']\!]_{\mathsf{SM}}$ where $\boldsymbol{P}' = \langle \mathcal{R}, \mathcal{P}_{j+1}, \ldots, \mathcal{P}_{n-1} \rangle$.*

Proof. See Appendix F.1, page 281. □

8.1.2 Semantic Properties

We proceed by examining further properties of rule update operators – of those generated by simple exception functions in general, and of the δ_{a}-, δ_{b}- and δ_{c}-based ones in particular. The properties we consider in this section are *semantic* in that they put conditions on the *models* of a result of an update and do not need to refer to the syntax of the original and updating programs. Our results are summarised in Table 8.1 and in the following we explain and discuss them. The interested reader may find all the proofs in Appendix F.1.5 starting on page 281.

Table 8.1: Semantic properties of simple rule update operators

Property	Formalisation	Type of $\equiv$, $\models$ and $[\![\cdot]\!]$							
		SU	RR	SR	RMR	SMR	RE	SE	SM
(Initialisation)	$\emptyset \oplus \mathcal{U} \equiv \mathcal{U}$.		✓	✓	✓	✓	✓	✓	✓
(Disjointness)	If $\mathcal{R}, \mathcal{S}$ are over disjoint alphabets, then $(\mathcal{R} \cup \mathcal{S}) \oplus \mathcal{U} \equiv (\mathcal{R} \oplus \mathcal{U}) \cup (\mathcal{S} \oplus \mathcal{U})$.		✓	✓	✓	✓	✓	✓	✓
(Non-interference)[†]	If $\mathcal{U}, \mathcal{V}$ are over disjoint alphabets, then $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V} \equiv (\mathcal{R} \oplus \mathcal{V}) \oplus \mathcal{U}$.		✓[abc]	✓[abc]	✓[abc]	✓[abc]	✓[abc]	✓[abc]	✓[abc]
(Tautology)	If $\mathcal{U}$ is tautological, then $\mathcal{R} \oplus \mathcal{U} \equiv \mathcal{R}$.		✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]
(Immunity to Tautologies)	If $\mathcal{S}$ and $\mathcal{V}$ are tautological, then $(\mathcal{R} \cup \mathcal{S}) \oplus (\mathcal{U} \cup \mathcal{V}) \equiv \mathcal{R} \oplus \mathcal{U}$.		✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]	✓[⋆]
(Idempotence)	$\mathcal{R} \oplus \mathcal{R} \equiv \mathcal{R}$.		✓[c]	✓[c]	✓	✓	✓	✓	✓
(Absorption)	$(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U} \equiv \mathcal{R} \oplus \mathcal{U}$.		✓[c]	✓[c]	✓[bc]	✓[bc]	✓[bc]	✓[bc]	✓[bc]
(Augmentation)[†]	If $\mathcal{U} \subseteq \mathcal{V}$, then $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V} \equiv \mathcal{R} \oplus \mathcal{V}$.		✓[c]	✓[c]	✓[bc]	✓[bc]	✓[bc]	✓[bc]	✓[bc]
(Associativity)	$\mathcal{R} \oplus (\mathcal{U} \oplus \mathcal{V}) \equiv (\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V}$.								
(P1)	$\mathcal{R} \oplus \mathcal{U} \models \mathcal{U}$		✓	✓	✓	✓	✓	✓	n/a
(P2.⊤)	$\mathcal{R} \oplus \emptyset \equiv \mathcal{R}$		✓	✓	✓	✓	✓	✓	✓
(P2.1)	$\mathcal{R} \cup \mathcal{U} \models \mathcal{R} \oplus \mathcal{U}$				✓	✓	✓	✓	n/a
(P2.2)	$(\mathcal{R} \cup \mathcal{U}) \oplus \mathcal{U} \models \mathcal{R}$								n/a
(P3)	If $[\![\mathcal{R}]\!] \neq \emptyset$ and $[\![\mathcal{U}]\!] \neq \emptyset$, then $[\![\mathcal{R} \oplus \mathcal{U}]\!] \neq \emptyset$	n/a	n/a	n/a	n/a	n/a			
(P4)	If $\mathcal{R} \equiv \mathcal{S}$ and $\mathcal{U} \equiv \mathcal{V}$, then $\mathcal{R} \oplus \mathcal{U} \equiv \mathcal{S} \oplus \mathcal{V}$		✓[⋆]						
(P5)	$(\mathcal{R} \oplus \mathcal{U}) \cup \mathcal{V} \models \mathcal{R} \oplus (\mathcal{U} \cup \mathcal{V})$				✓	✓	✓	✓	n/a
(P6)	If $\mathcal{R} \oplus \mathcal{U} \models \mathcal{V}$ and $\mathcal{R} \oplus \mathcal{V} \models \mathcal{U}$, then $\mathcal{R} \oplus \mathcal{U} \equiv \mathcal{R} \oplus \mathcal{V}$								n/a

[a] Holds if $\oplus$ is generated by δ_a.
[b] Holds if $\oplus$ is generated by δ_b.
[c] Holds if $\oplus$ is generated by δ_c.
[⋆] Holds if $\oplus$ is generated by a local exception function δ such that $\delta(\mathcal{M}, \mathcal{X}) \subseteq \mathcal{M}$ for all $\mathcal{M} \subseteq \mathcal{X}$. This is satisfied by δ_a, δ_b and δ_c.
[†] Results on this line hold only if $\mathcal{R}, \mathcal{U}, \mathcal{V}$ are non-disjunctive programs.

Semantic Properties of Rule Updates

The properties in the upper part of Table 8.1 were introduced in (Eiter et al., 2002; Alferes et al., 2005; Delgrande et al., 2007). All of them are formalised for rule bases $\mathcal{R}$, $\mathcal{S}$, $\mathcal{U}$, $\mathcal{V}$ and a rule update operator $\oplus$. Each of them can actually be seen as a *meta-property* that is instantiated once we adopt a particular notion of program equivalence. Therefore, each row of Table 8.1 has eight cells that stand for particular instantiations of the property. This provides a more complete picture of how simple rule update operators, properties and program equivalence are interrelated.

Unless stated otherwise (in a footnote), each tick (✓) signifies that the property in question holds for *all* simple rule update operators. A missing tick signifies that the property does not generally hold for simple rule update operators, and in particular there are δ_a-, δ_b- and δ_c-based operators for which it is violated. A tick is smaller if it is a direct consequence of a preceding larger tick in the same row and of the interrelations between the program equivalence notions (c.f. Figure 7.1).

At a first glance, it is obvious that none of the semantic properties is satisfied under SU-equivalence. This is because the conditions placed on a rule update operator by an exception function are at the semantic level, while SU-equivalence effectively compares programs syntactically. For instance, an exception-based operator $\oplus$, for any exception function ε, may behave as follows: $\emptyset \oplus \{ \sim p \leftarrow p. \} = \{ \leftarrow p. \}$. This is because the rules before and after update are RE-equivalent. However, due to the fact that the programs $\{ \sim p \leftarrow p. \}$ and $\{ \leftarrow p. \}$ are considered to be different under SU-equivalence, $\oplus$ cannot satisfy (Initialisation) w.r.t. SU-equivalence. The situation with all other properties is analogous.

In the following we discuss the properties in the upper part of Table 8.1 w.r.t. the remaining notions of equivalence.

(Initialisation) and (Disjointness): These properties are satisfied "by construction", regardless of which simple rule update operator we consider and of which notion of equivalence we pick.

(Tautology) and (Immunity to Tautologies): These are naturally satisfied by all simple update operators that do not introduce exceptions merely due to the presence of a tautological rule in the updating program. In particular, both properties are satisfied by δ_a-, δ_b- and δ_c-based operators. Note that these properties are generally acknowledged as very desirable although most existing rule update semantics fail to comply with them.

(Non-interference): This property is not guaranteed for arbitrary simple update operators, but it is satisfied if we constrain ourselves to δ_a-, δ_b- or δ_c-based ones. However, this is only the case when non-disjunctive programs are considered. This points towards one of the important open problems faced by state-of-the-art research on rule updates: examples, desirable properties and methods for updating *disjunctive* programs. Insights in this direction should shed light on whether (Non-interference) is desirable in the disjunctive case.

(Idempotence), (Absorption) and (Augmentation): These are the only properties that reveal differences amongst δ_a, δ_b and δ_c. They are not satisfied by δ_a- and δ_b-based operators under SR- and RR-equivalence. The reason for this is that when a program is updated by its subset, exceptions may still be introduced to some rules, resulting in weakened versions of the original rules. Since such rules are not part of the original program, the programs before and after update are considered to be different under SR- and RR-equivalence. This problem is dodged in δ_c by completely eliminating original rules that appear in the update. This also seems to indicate that SR- and RR-equivalence are slightly *too strong* for characterising updates because programs such as $\{ p. \}$ and $\{ p., p \leftarrow q. \}$ are not considered equivalent even though we expect the same behaviour from them when they are updated. We speculated in Chapter 7 that this could be solved by adopting the weaker SMR- and RMR-equivalence. However, it turns out these equivalence relations are *too weak*: programs such as $\{ \sim p. \}$ and $\{ \sim p., q \leftarrow p. \}$ are SMR- and RMR-equivalent although, when updated by $\{ p. \}$, different results are expected for each of them.

Moreover, δ_a-based operators fail to satisfy (Absorption) and (Augmentation). Along with Theorem 8.6, this seems to indicate that δ_a does not correctly handle iterated updates.

(Associativity): This is one of the few properties that is not satisfied by any of the defined classes of operators. This is closely related to the question of whether *rejected rules are allowed to reject*. (Associativity) can be seen as postulating that an update operator must behave the same way regardless of whether rejected rules are allowed to reject or not. As witnessed by the AS- and JU-semantics (c.f. equation 2.2), rule update semantics tend to generate unwanted models when rejected rules are not allowed to reject.

Reformulation of Belief Update Postulates

The lower part of Table 8.1 contains a straightforward reformulation of the first six belief update postulates for rule bases. We omit the last two postulates as they require program disjunction and it is not clear how to obtain it appropriately. Note also that (B7) has been heavily criticised in the literature as being mainly a means to achieve formal results instead of an intuitive principle (Herzig and Rifi, 1999) and though (B8) reflects the basic intuition behind belief update – that of updating each model independently of the others – such a point of view is hardly transferable to knowledge represented using rules because a single model, be it a classical, stable, SE- or RE-model, fails to encode the interdependencies between literals expressed in rules that are necessary for properties such as support.

Since we did not define SM-entailment, postulates that refer to it have the SM column marked as "n/a".

We now turn to the individual postulates.

(P1) and (P2.$\top$): Similarly as (Initialisation) and (Disjointness), these postulates are satisfied by any simple rule update operator and under all notions of equivalence.

(P2.1) and (P5): Postulate (P2.1) is not satisfied under SR- and RR-equivalence for the same reasons, described above, that prevent δ_a- and δ_b-based operators from satisfying (Idempotence). The situation with (P5) is the same since it implies (P2.1) in the presence of (P2.$\top$).

(P2.2) and (P6): Postulate (P2.2) requires that $\{ p., \sim p. \} \oplus \{ \sim p. \} \models p$ which, in the presence of (P1), amounts to postulating that one can never recover from an inconsistent state, contrary to most rule update semantics which do allow for recovery from such states. The case of (P6) is the same since it implies (P2.2) in the presence of (P1) and (P2.$\top$).

(P3): This postulate relies on a function that returns the set of models of a rule base. Thus, $[\![\cdot]\!]_{\mathsf{SM}}$, $[\![\cdot]\!]_{\mathsf{SE}}$ and $[\![\cdot]\!]_{\mathsf{RE}}$ can be used for this purpose and the other columns in the corresponding row in Table 8.1 make little sense, so they are marked as "n/a". Furthermore, this postulate is not satisfied by any of the defined classes of exception-based operators. It is also one of the principles that most existing approaches to rule update chronically fail to satisfy. In order to satisfy it, a context-aware exception function would have to be used because conflicts may arise in a set of more than two rules that are pairwise consistent. For instance, when updating $\{\, p. \,\}$ by $\{\, q \leftarrow p., {\sim}q \leftarrow p. \,\}$, one would somehow need to detect and resolve the joint conflict between these three rules. This is however impossible with a simple exception function because it only considers conflicts between pairs of rules, one from the original program and one from the update.

(P4): This postulate requires update operators to be syntax-independent. In this context it is useful to also consider the following related principles, derived from (B4.1), (B4.2) and (B8.2):

(P4.1) If $\mathcal{P} \equiv \mathcal{Q}$, then $\mathcal{P} \diamond \mathcal{U} \equiv \mathcal{Q} \diamond \mathcal{U}$.

(P4.2) If $\mathcal{U} \equiv \mathcal{V}$, then $\mathcal{P} \diamond \mathcal{U} \equiv \mathcal{P} \diamond \mathcal{V}$.

(P8.2) If $\mathcal{P} \models \mathcal{Q}$, then $\mathcal{P} \diamond \mathcal{U} \models \mathcal{Q} \diamond \mathcal{U}$.

Following the arguments from Chapter 6, the failure to satisfy (P4.1) under SM-, SE- and RE-equivalence is inevitable if properties such as support and fact update are to be respected. Furthermore, programs such as $\mathcal{P}_1 = \{\, {\sim}p. \,\}$ and $\mathcal{P}_2 = \{\, {\sim}p., q \leftarrow p. \,\}$ are SMR- and RMR-equivalent even though an update by $\mathcal{U} = \{\, p. \,\}$ is expected to produce different results when applied to them. Indeed, the JU-semantics assigns the stable model $\{\, p \,\}$ to the DLP $\langle \mathcal{P}_1, \mathcal{U} \rangle$ and the stable model $\{\, p, q \,\}$ to the DLP $\langle \mathcal{P}_2, \mathcal{U} \rangle$. It thus follows from Theorems 8.6 and 8.9 that (P4.1) is not satisfied under RMR- and SMR-equivalence (nor any other weaker notion of equivalence) by operators generated by exception functions δ_{a}, δ_{b} and δ_{c}. Under SR-equivalence, (P4.1) is also violated due to the fact that a constraint such as $(\leftarrow p.)$ cannot be weakened by the introduced exception functions while the fact $({\sim}p.)$ can, although it is strongly equivalent to the constraint. Moreover, since (P8.2) is a stronger principle than (P4.1), these counterexamples apply to it as well.

Similar arguments can also be used to show that the principle (P4.2) is not satisfied under SM-, SE-, RE, SMR- and RMR-equivalence. We only need to observe that any δ_{a}-, δ_{b}- or δ_{c}-based operator $\oplus$ satisfies $\emptyset \oplus \mathcal{P}_1 \equiv_{\mathsf{RR}} \mathcal{P}_1$ and $\emptyset \oplus \mathcal{P}_2 \equiv_{\mathsf{RR}} \mathcal{P}_2$, and thus by the previous reasoning, $\emptyset \oplus \mathcal{P}_1 \oplus \mathcal{U}$ has different stable models than $\emptyset \oplus \mathcal{P}_2 \oplus \mathcal{U}$. This contradicts (P4.2) under RMR-equivalence and any other weaker equivalence as well. Additionally (P4.2) is not satisfied under SR-equivalence due to the fact that updates such as $\{\, {\sim}p \leftarrow q. \,\}$, $\{\, {\sim}q \leftarrow p. \,\}$ and $\{\, \leftarrow p, q. \,\}$ have different effects on the program $\{\, p., q. \,\}$.

These observations seem to indicate that (P4.1), (P4.2), (P8.2), and thus also (P4), are too strong under SM-, SE-, RE-, SMR- and RMR-equivalence. Under SR-equivalence, they are incompatible with operators that solve conflicts based on heads of rules. On the other hand, due to the semantic underpinning of simple rule update operators, (P4) is satisfied by most of them, including all δ_{a}-, δ_{b}- and δ_{c}-based ones, under RR-equivalence. It might be interesting to look for constrained classes of exception functions that satisfy syntax-independence w.r.t. SR-equivalence. Such functions, however, will not be able to respect causal rejection because SE-models cannot distinguish abolishing rules.

8.2 General-Purpose Exception-Based Operators

The ideas behind exception-based operators are not limited to dealing with rule updates. In the present section we formulate them abstractly, for any knowledge representation formalism with a monotonic model-theoretic semantics. Subsequently, we show that the abstract framework can be instantiated to the case of first-order logic and is able to capture a range of model-based as well as formula-based update operators.

Throughout this section we assume to be using some knowledge representation formalism in which a *knowledge base* is a subset of the set of all *knowledge atoms* Ω and $\mathcal{Z}$ denotes the set of all *semantic structures* among which the *models* of knowledge atoms are chosen. The set of models of a knowledge atom α is denoted by $[\![\alpha]\!]$. The *semantic characterisation* of a knowledge base $\mathcal{K}$ is the *set of sets of models* of its knowledge atoms: $\langle\!\langle \mathcal{K} \rangle\!\rangle = \{ [\![\alpha]\!] \mid \alpha \in \mathcal{K} \}$. The models of $\mathcal{K}$ are the models of all its elements, i.e. $[\![\mathcal{K}]\!] = \bigcap \langle\!\langle \mathcal{K} \rangle\!\rangle$.

As in the case of rule updates, an exception-based operator views a knowledge base $\mathcal{K}$ through its semantic characterisation $\langle\!\langle \mathcal{K} \rangle\!\rangle$ and *introduces exceptions* to its knowledge atoms by adding new semantic structures to their original sets of models. The formalisation of this idea is straight-forward: an exception-based update operator is characterised by an *exception function* that, given the set of models of a knowledge atom α and the semantic characterisations of the original and updating knowledge base, returns the set of semantic structures that are to be introduced as exceptions to α.

Definition 8.12 (Exception Function)**.** An *exception function* is any function

$$\varepsilon : 2^{\mathcal{Z}} \times 2^{2^{\mathcal{Z}}} \times 2^{2^{\mathcal{Z}}} \rightarrow 2^{\mathcal{Z}} \enspace .$$

Given such an exception function and knowledge bases $\mathcal{K}$, $\mathcal{U}$, it naturally follows that the semantic characterisation resulting from updating $\mathcal{K}$ by $\mathcal{U}$ should consist of sets of models of each knowledge atom α from $\mathcal{K}$, each augmented with the respective exceptions, and also the unmodified sets of models of knowledge atoms from $\mathcal{U}$. In other words, we obtain the set of sets of models

$$\{ [\![\alpha]\!] \cup \varepsilon([\![\alpha]\!], \langle\!\langle \mathcal{K} \rangle\!\rangle, \langle\!\langle \mathcal{U} \rangle\!\rangle) \mid \alpha \in \mathcal{K} \} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle \enspace . \tag{8.3}$$

Turning to the syntactic side, an *update operator* is binary function over 2^{Ω} that takes the original knowledge base and its update as inputs and returns the updated knowledge base. An *exception-based update operator* is then formalised as follows:

Definition 8.13 (Exception-Based Update Operator)**.** We say that an update operator $\oplus$ is *exception-based* if for some exception function ε, $\langle\!\langle \mathcal{K} \oplus \mathcal{U} \rangle\!\rangle$ is equal to (8.3) for all $\mathcal{K}, \mathcal{U} \subseteq \Omega$. In that case we also say that $\oplus$ is *ε-based*.

8.3 Belief Updates Using Exception-Based Operators

The introduced abstract framework can be instantiated for various formalisms. Previously we considered exception-based rule update operators which correspond to the case when Ω is the set of all rules, $\mathcal{Z}$ is the set of all three-valued interpretations $\mathfrak{X}$ and the semantic function $[\![\cdot]\!]$ returns the RE-models of its argument.

Concrete exception-based operators for first-order theories are obtained from the abstract framework by identifying the set of knowledge atoms Ω with the set of first-order sentences and the set of semantic structures $\mathcal{Z}$ with first-order interpretations under the standard names assumption $\mathfrak{I}$. The semantic function $[\![\cdot]\!]$ returns the models of the argument sentence or theory.

As shown in Section 2.6, model-based update operators, such as Winslett's, satisfy principles (FO1) – (FO6) and (FO8.2). The following result shows that all operators that satisfy (FO1), (FO2.1) and (FO4), including all model-based update operators, can be faithfully modelled by an exception function.

Theorem 8.14 (Model-Based Updates Using Exception-Based Operators)**.** *If $\diamond$ is an update operator that satisfies (FO1), (FO2.1) and (FO4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of theories $\boldsymbol{T}$, $[\![\diamond \boldsymbol{T}]\!] = [\![\bigoplus \boldsymbol{T}]\!]$.*

Proof. See Appendix F, page 289. □

This essentially means that any operator that satisfies *primacy of new information*, formalised in (FO1), retains models of the initial theory that are also models of the update, expressed in (FO2.1), and is syntax-independent, as captured by (FO4), can be equivalently cast into the framework of exception-based updates.

Similar results can be achieved for formula-based update operators. First we introduce the following principles, counterparts of the respective belief update postulates, which are satisfied by many formula-based operators. We denote by $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}}$ the set $\langle\!\langle \mathcal{T} \rangle\!\rangle \cup \{ \mathfrak{I} \}$ for any theory $\mathcal{T}$.[3] The principles are as follows:

(F1) $\langle\!\langle \mathcal{T} \circ \mathcal{U} \rangle\!\rangle \supseteq \langle\!\langle \mathcal{U} \rangle\!\rangle$.

(F2.1) $\langle\!\langle \mathcal{T} \cup \mathcal{U} \rangle\!\rangle \supseteq \langle\!\langle \mathcal{T} \circ \mathcal{U} \rangle\!\rangle$.

(F4) If $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$, then $\langle\!\langle \mathcal{T} \circ \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \circ \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$.

We can see that (F1) and (F2.1) are *stronger* versions of (FO1), and (FO2.1), respectively. While (F1) requires that the sets of models of formulae in $\mathcal{U}$ be retained in the semantic characterisation of $\mathcal{T} \circ \mathcal{U}$, (F2.1) states that every formula in $\mathcal{T} \circ \mathcal{U}$ be equivalent to some formula in $\mathcal{T} \cup \mathcal{U}$. Intuitively, this means that $\mathcal{T} \circ \mathcal{U}$ is obtained from $\mathcal{T} \cup \mathcal{U}$ by deleting some of its elements, modulo equivalence. Finally, (F4) is a reformulation of (FO4) – it can be seen as syntax-independence w.r.t. the set of sets of models of a first-order theory, modulo the presence of tautologies, instead of the overall set of models as in (FO4). In some ways it is *weaker* than (FO4) as its antecedent is much stronger.

The WIDTIO, Cross-Product and Bold operators, introduced in Definitions 2.32, 2.33 and 2.48, respectively, can be straight-forwardly generalised to deal with first-order theories.[4] The WIDTIO operator satisfies principles (F1), (F2.1) and (F4), and so does the Bold operator if it is based on a remainder selection function that selects remainders with the same semantic characterisation when given sets of remainders with the same sets of semantic characterisations. More formally:

Definition 8.15 (Regular Bold Operator)**.** Let $\mathcal{R}$ be a set of remainders. We denote the set $\{ \langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} \mid \mathcal{T}' \in \mathcal{R} \}$ by $(\!(\mathcal{R})\!)^{\mathfrak{I}}$.

We say that the Bold operator $\circ^{s}_{\text{BOLD}}$ is *regular* if for all sets of remainders $\mathcal{R}_1$, $\mathcal{R}_2$ such that $(\!(\mathcal{R}_1)\!)^{\mathfrak{I}} = (\!(\mathcal{R}_2)\!)^{\mathfrak{I}}$ it holds that $\langle\!\langle s(\mathcal{R}_1) \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle s(\mathcal{R}_2) \rangle\!\rangle^{\mathfrak{I}}$.

[3] Recall that $\mathfrak{I}$ denotes the set of all two-valued interpretations.

[4] In case of the Cross-Product operator this generalisation requires that theories be finite.

The regularity condition guarantees a certain degree of independence of syntax, e.g. given the sets of remainders $\mathcal{R}_1 = \{\{p\}, \{q\}\}$ and $\mathcal{R}_2 = \{\{p \wedge p\}, \{q \vee q\}\}$, a regular Bold operator either selects $\{p\}$ from $\mathcal{R}_1$ and $\{p \wedge p\}$ from $\mathcal{R}_2$, or it selects $\{q\}$ from $\mathcal{R}_1$ and $\{q \vee q\}$ from $\mathcal{R}_2$. A non-regular one might select, say, $\{p\}$ from $\mathcal{R}_1$ and $\{q \vee q\}$ from $\mathcal{R}_2$. Thus the regularity condition ensures that the operator is independent of the syntax of individual sentences in the first-order theory.

The Cross-Product operator satisfies (F1), (FO2.1) and (F4), but not (F2.1).

Proposition 8.16 (Properties of Formula-Based Updates)**.** *The WIDTIO and regular Bold operators satisfy (F1), (F2.1) and (F4). The Cross-Product operator satisfies (F1), (FO2.1) and (F4) but does not satisfy (F2.1).*

Proof. See Appendix F, page 294. □

The following result establishes that formula-based operators such as WIDTIO and regular Bold can be fully captured by exception-based operators. In addition, operators such as Cross-Product can be captured for the case of a single update.

Theorem 8.17 (Formula-Based Updates Using Exception-Based Operators)**.** *If $\circ$ is an update operator that satisfies (F1), (F2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of theories $\boldsymbol{T}$, $[\![\bigcirc \boldsymbol{T}]\!] = [\![\bigoplus \boldsymbol{T}]\!]$.*

If $\circ$ is an update operator that satisfies (F1), (FO2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all theories $\mathcal{T}, \mathcal{U}$, $[\![\mathcal{T} \circ \mathcal{U}]\!] = [\![\mathcal{T} \oplus \mathcal{U}]\!]$.

Proof. See Appendix F, page 296. □

An interesting point regarding the obtained results is that the principles (FO1), (FO2.1) and (FO4) are not specific to update operators, they are also satisfied by *AGM revision* operators. These operators are developed for the case of revising a *belief set* which is a set of formulae closed w.r.t. a logical consequence operator Cn. A revision operator $\star$ takes an original belief set $\mathcal{T}$ and a formula μ, representing its revision, and produces the revised belief set $\mathcal{T} \star \mu$. The properties satisfied by AGM revision operators include *success*, *inclusion* and *extensionality* (Hansson, 1993a), formalised, respectively, as

$$\mu \in \mathcal{T} \star \mu \ , \qquad \mathcal{T} \star \mu \subseteq Cn(\mathcal{T} \cup \{\mu\}) \ , \qquad \text{If } \mu \equiv \nu, \text{ then } \mathcal{T} \star \mu = \mathcal{T} \star \nu.$$

These three properties directly imply that (FO1), (FO2.1) and (FO4) are satisfied by AGM revision operators if the initial knowledge base is a belief set and each of its updates a single formula. This essentially means that Theorem 8.14 directly applies to AGM revision operators as well. Note that the operator adopted for ABox updates in (Lenzerini and Savo, 2011), inspired by WIDTIO, performs a deductive closure of the ABox before updating it, so it corresponds to the standard *full meet AGM revision operator*.

Similarly, principles (F1), (F2.1) and (F4) are closely related to the properties of *base revision operators* (Hansson, 1993a), of which direct instances are the WIDTIO and Bold operators. In particular, two types of base revision are identified in (Hansson, 1993a), the *internal* and *external base revision*. Both of them satisfy base revision counterparts of *success* and *inclusion* and, in addition, internal revision operators satisfy a property called *uniformity*:

(Uniformity) If, for all $\mathcal{B}' \subseteq \mathcal{B}$, $\mathcal{B}' \cup \mathcal{U}$ is inconsistent iff $\mathcal{B}' \cup \mathcal{V}$ is inconsistent, then $\mathcal{B} \cap (\mathcal{B} \star \mathcal{U}) = \mathcal{B} \cap (\mathcal{B} \star \mathcal{V})$.

These three principles together entail that internal revision operators satisfy (F1), (F2.1) and one half of (F4); the other half can be achieved by putting additional constraints on the two-place selection function that generates the revision operator, similar to the *regularity* condition we imposed on the Bold operator above. Such regular internal revision operators are thus directly subject to Theorem 8.17. The same however does not hold for regular *external* revision operators as they need not satisfy *uniformity*. Note also that the WIDTIO and Bold operators coincide with *internal full meet base revision* and *internal maxichoice base revision* operators, respectively.

8.4 Discussion

We have introduced *exception-based operators* which view a theory or program as the *set of sets of models* of its elements, and perform updates by adding new interpretations – *exceptions* – to the sets of models of elements in the original theory or program.

The most important feature of this approach is that it provides a common basis for a wide range of update semantics. On the one hand, we have shown that it can characterise the JU-semantics, a traditional syntax-based approach to rule updates. On the other hand, it can fully capture update operators that form the basis of ontology updates (Liu et al., 2006; De Giacomo et al., 2009; Calvanese et al., 2010; Lenzerini and Savo, 2011), such as the model-based Winslett's operator, or the formula-based WIDTIO and Bold operators. In addition, the Cross-Product operator can be captured for the case of performing a single update and the same can be said about the Set-Of-Theories approach (Fagin et al., 1983) since for a single update it is equivalent to the Cross-Product operator (Winslett, 1990). However, these two operators do not offer a viable alternative for updating ontologies. Cross-Product requires that disjunctions of ontology axioms be performed, which is typically not supported in DLs, and Set-Of-Theories produces a disjunctive ontology which is impractical and deviates from mainstream DL research.

The newly found bindings between belief and rule updates can already be observed by looking at our investigation of semantic properties of simple exception-based rule update operators. Previously, the reformulation of belief update postulates and their study in the context of rule updates was problematic because many mature rule update semantics define only the stable models of a DLP and do not construct the updated program. By finding an actual exception-based operator that produces the new program, we are able to look at belief update postulates from various perspectives and analyse different notions of program equivalence and entailment w.r.t. particular operators and postulates (c.f. Table 8.1).

Overall, exception functions and exception-based operators offer a uniform framework that bridges two very distinct approaches to updates, previously considered irreconcilable. The relationship between exception-based operators and revision operators, on both belief sets and belief bases, may bring new insights into the approaches to *ontology revision* (Qi and Yang, 2008; Qi et al., 2008; Halaschek-Wiener and Katz, 2006; Ribeiro and Wassermann, 2007). This also seems relevant even in the context of ontology updates since it has been argued in the literature that the strict distinction between revision and update is not suitable in the context of ontologies (Calvanese et al., 2010).

Part IV

A Glance Astern and Ahead

9

Conclusions and Future Directions

I may not have gone where I intended to go, but I think I have ended up where I needed to be.

Douglas Adams
English humorist & science fiction novelist

This final chapter contains a summary of our main contributions, viewing them from a broader perspective, and discusses interesting lines of future research.

In the previous parts of the thesis we presented a number of results that bring new insights regarding state-of-the-art approaches to knowledge updates. Part II introduces and examines the first *update semantics for hybrid knowledge bases* and pinpoints the obstacles in defining a universal hybrid update semantics. Part III seeks to find *semantic characterisations of rule updates* in order to bring them closer to ontology updates. Both research directions have previously been largely unexplored and our investigation illustrates the extent to which ontology and rule updates can be naturally integrated with one another, points out their distinguishing properties, and introduces a unified perspective that can capture both these distinct update paradigms.

In the following sections we take a closer look at the conclusions drawn from our results. Then we point at desirable future developments.

9.1 Updates of Hybrid Knowledge Bases

In Chapters 3 and 4 we defined two update semantics for MKNF knowledge bases, a mature framework for representing and reasoning with tightly integrated hybrid knowledge. The first semantics uses a first-order update operator $\diamond$ to perform ontology updates in the presence of static rules. It encompasses applications of hybrid knowledge bases in which the ontology contains highly dynamic information while rules represent defaults, preferences or behaviour that does not undergo changes and can be overridden by ontology updates when necessary.

The second semantics modularly combines a first-order update operator $\diamond$ with a rule update semantics S in order to update MKNF knowledge bases consisting of ontology and rule layers that share information through a rule-based interface. We demonstrated that it is useful in a realistic scenario as it is capable of performing non-trivial updates, automatically resolving conflicts in the expected manner, and propagating new information across the knowledge base.

The two hybrid update semantics are complementary in the sense that each can handle inputs that the other one cannot. Moreover, they are fully compatible with one another, meaning that they assign the same semantics to inputs accepted by both (c.f. Theorem 4.34). By combining them one can thus obtain an integrated hybrid update semantics that can safely handle inputs treated by either of the two semantics.

In addition, we make only minimal assumptions about the properties of $\diamond$ and S that guarantee the correctness of our definitions. In other words, we abstract away from particular instantiations of $\diamond$ and S and allow for any approach to ontology and rule updates, be it an existing one or one that is yet to be discovered, as long as it satisfies those assumptions. The introduced semantics thus concentrate on the interplay between ontology and rules and can serve as points of comparison for other hybrid update semantics that might be developed in the future.

We also examined the basic theoretical properties of our hybrid update semantics. We showed that they are faithful to $\diamond$ and S, i.e. they preserve their behaviour if the hybrid knowledge base contains only ontology axioms or only rules. Similarly, the semantics are faithful to the static semantics of MKNF knowledge bases, so when no updates are performed, the assigned models are simply MKNF models of the initial MKNF knowledge base. Furthermore, they respect the principle of primacy of new information.

The practical usefulness of the introduced semantics is underlined by the fact that the full expressivity of MKNF knowledge bases does not seem to be necessary in a number of use cases of hybrid knowledge. Especially, the separation of a hybrid knowledge base into distinct ontology and rule layers, as required by the semantics from Chapter 4, seems to be a natural way of controlling, from the perspective of a knowledge engineer, how the different types of knowledge interact.

Having established these positive results, in Chapter 5 we investigated the possibility of defining a universal hybrid update semantics. However, this led to the identification of a number of serious obstacles. First, we showed that the traditional model-based approach to updates leads to counterintuitive results when used for updating TBoxes and we formally pinpointed this observation. Its main consequence in the context of hybrid knowledge bases is that one can hardly propose a plausible hybrid update semantics when it is not clear how the intuitions regarding TBox updates can be reconciled with model-based update operators that underlie ABox updates.

Even more importantly, we demonstrated the clash of intuitions due to the incompatibilities between principles underlying belief and rule updates. These observations lead to the conclusion that without a deeper understanding of the distinct approaches to updates and of their mutual relation, a plausible universal hybrid update semantics can hardly be defined. Furthermore, since rule update semantics rely on the syntactic structure of rules, which is absent in ontologies, this motivated us to look for semantic characterisations of rule updates that would bridge ontology updates with rule updates, which we tackled in Part III.

9.2 Semantic Characterisations of Rule Updates

We commenced our investigation of semantic rule update operators in Chapter 6 by casting Katsuno and Mendelzon's belief update framework to logic program updates, with each program characterised by the set of its SE-models. We proved a representation theorem that shows a one-to-one relationship between operators that satisfy the reformulated belief update postulates and operators constructively defined using preorder assignments, similarly as in the case of belief update operators. We also defined a counterpart of Winslett's update operator for performing program updates which satisfies the reformulated postulates.

However, after a closer investigation we found out that all operators that satisfy the syntax-independence postulate $(\text{P4})_{\text{SE}}$ violate either *support* or *fact update*, both of which are basic and desirable properties that are generally satisfied by existing rule update semantics. In other words, program updates based on SE-models and belief update postulates are incompatible with traditional syntax-based approaches to rule updates. The main reason for this is that the set of SE-models of a program does not capture the literal dependencies expressed by its rules.

This led us to the search for more expressive semantic characterisations of logic programs. In Chapter 7 we proposed to view a program as the *set of sets of models of its rules*. We examined the rule equivalence classes induced by SE-models and learned that even though SE-models are capable of distinguishing most rules that are treated differently by traditional rule update semantics, they fail to differentiate between rules with default literals in their heads and constraints, which carry different meaning in the context of causal rejection-based rule update semantics. To overcome this limitation, we introduced RE-models and proved that they are able to distinguish the desired classes of rules while retaining the essential properties of SE-models. We also introduced notions of program equivalence and entailment based on viewing the program as the set of sets of SE- and RE-models of its rules, and compared them in terms of strength.

Subsequently, in Chapter 8, we proposed a generic method for defining semantic rule update operators: a program, viewed as the *set of sets of RE-models of its rules*, is updated by introducing additional interpretations – *exceptions* – to the original sets of RE-models. Using this approach we were able to arrive at a class of semantic rule update operators that are tightly related to the JU-semantics for rule updates. This constitutes an important advancement since it bridges, for the first time, a semantic approach to rule updates with a syntactic one.

Furthermore, the introduced exception-based operators do not only assign a set of stable models to a sequence of programs, as is the case with the JU-semantics and other traditional rule update semantics, but produce an actual syntactic object that represents the updated program. In other words, they compile any sequence of programs to a single rule base. This brought about a new insight into the problem of *state condensing* from rule update literature. In addition, it allowed us to extensively examine the semantic properties of rule updates under different notions of equivalence and entailment, providing a much broader picture of properties of rule updates than the one attainable previously. In particular, the operators naturally satisfy the syntax-independence postulate (P4) under RR-equivalence, which views a program through the set of sets of RE-models of its rules, and enjoy a range of other desirable properties as well.

Moreover, we showed that exception-based operators can also capture a range of *belief update operators*. This means that they provide a unified framework under which the two different update paradigms can be further investigated.

9.3 Future Directions

The results of this thesis encourage a further investigation in a number of different directions, summarised as follows:

Further development of proposed hybrid update semantics: The update semantics for MKNF knowledge bases from Chapters 3 and 4 can be further generalised by lifting some the constraints under which they are developed. In case of the former semantics, it would be interesting to consider adding support for disjunctive rules and for updates of the rule component, even if only to a limited extent.

As for the latter semantics, one of the issues to study are the splitting properties that it is based on. The main limitation is that the splitting sets contain *predicate symbols* while in some cases it would be desirable to allow for a more fine-grained splitting, on the level of *ground atoms*. A related and seemingly more demanding problem is the treatment of the equality predicate $\approx$: How can the semantics be extended to account for the presence of equality assertions and TBox axioms whose translation to first-order logic requires equality, such as number restrictions?

Furthermore, we have shown that the splitting properties are not satisfied by all ontology and rule update semantics, raising questions as to whether they are perhaps *too strong*. At the first glance it would seem that the splitting properties are not strong at all since their only requirement is that syntactically independent parts of a knowledge base also be semantically independent. However, this only seems to be a reasonable requirement if we are interested in finding a *domain-independent update semantics*, as we typically are in the context of the Semantic Web. But if we are also interested in domain-dependent semantics, syntactic independence need not imply semantic independence. This also explains why many first-order update operators characterised by order assignments do not satisfy the splitting properties (c.f. Example 4.17) – if the order assignment encodes domain-dependent knowledge about how propositions ought to be updated, then despite the syntactic independence of two parts of a knowledge base, an update of one of them may trigger changes in the other one. Hence, a more general formulation of the splitting properties that accounts for these cases would be worth examining.

There is also an interesting relationship between the defined class of layered DMKBs and multi-context systems of Brewka and Eiter (2007). Each layer of a DMKB w.r.t. a particular layering splitting sequence can be viewed as a context that includes its own bridge rules. At the same time, the constraints we impose guarantee that each such context either contains only rules, so the context logic can be the stable models semantics, or it contains only DL axioms so that first-order logic can be used as its underlying logic. Though different splitting sequences induce different multi-context systems, their overall semantics remains the same (c.f. Proposition 4.27). A further study of this close relationship may bring about interesting new insights.

Finally, the development of similar update semantics under other existing frameworks for hybrid knowledge bases constitutes another topic of investigation.

Further development of semantic characterisations of rule updates: The developments of Chapters 6, 7 and 8 encourage further investigation along the following directions:

- Adding support for strong negation.

- Finding constructive syntactic definition of the introduced exception-based operators that would reveal more about the inner workings of the JU-semantics.
- Developing exception-based characterisations of other rule update semantics. This poses a number of challenges due to the need to detect non-tautological irrelevant updates (Alferes et al., 2005; Šefránek, 2006, 2011). For instance, simple exception functions examined in this thesis cannot distinguish an update of $\{ p. \}$ by $\mathcal{U} = \{ \sim p \leftarrow \sim q., \sim q \leftarrow \sim p. \}$, where it is plausible to introduce the exception $(\emptyset, \emptyset)$, from an update of $\{ p., q. \}$ by $\mathcal{U}$, where such an exception should not be introduced due to the cyclic dependency of justifications to reject facts $(p.)$ and $(q.)$. In such situations, context-aware exception functions need to be used. On the other hand, such functions have the potential for satisfying properties such as (Associativity) and (P3).
- Using the insights gained by exception-based characterisations of various rule update semantics to also shed light on the problem of *updating disjunctive programs* which has been given very little attention up until now.
- Looking for a notion of program equivalence that is stronger than RMR-equivalence and weaker than RR-equivalence and forms a suitable basis for rule updates, so that properties such as (P4) and (P2.1) can be achieved under a single notion of program equivalence. In particular, the results presented in Table 8.1 suggest that while RR-equivalence is too strong for properties such as (P2.1), RMR-equivalence is too weak for (P4).
- The possibility of characterising both belief and rule updates using exception functions has great potential in the context of updates of hybrid knowledge bases.
  The clash of intuitions regarding ABox, TBox and rule updates, presented in Chapter 5, can be seen in new light by studying the exception functions that lead to it. When coupled with a counterpart of SE- and RE-models in the context of MKNF knowledge bases, this can lead to universal hybrid update semantics which in turn improve our understanding of the relation between the distinct update paradigms.

Postulates for rule updates and hybrid updates: To this date, there is no generally accepted rule update semantics or a definitive set of properties, akin to KM postulates for belief update, that would reliably distinguish good rule update semantics from bad ones (c.f. (Eiter et al., 2002)). The search for formally formulated desirable properties of rule update and hybrid update semantics is a challenging and important future research area. Within this context, the properties for iterated revision (c.f. (Darwiche and Pearl, 1997)) should be further investigated in the context of updates.

TBox updates: TBox updates stand out as a particular area where novel ideas and solutions are necessary. As shown in Chapter 5, model-based update operators are not appropriate for updating TBoxes and the solution proposed by Calvanese et al. (2010) is based on ideas from belief *revision*. The main question that arises is whether and how the distinction between revision and updates applies to the evolution of TBoxes. Advances in this respect would also shed light on the general relationship between revision and updates and possibly also on ways of integrating them.

Computational properties: Computational complexity and algorithms for implementing the suggested hybrid update semantics and exception-based operators have not been addressed in this work and should be investigated in future research.

In this context, it would also be interesting to consider tractable approximations of the proposed semantics. The well-founded semantics for logic programs (Gelder et al., 1991) and its version for MKNF knowledge bases (Knorr et al., 2011) constitute crucial starting points in this direction.

Other change operations: Other change operations, such as *forgetting*, *erasure*, *revision* and *contraction*, also need to be studied and related to updates in the context of hybrid knowledge bases.

Overall, our results show how updates within interesting use cases of hybrid knowledge can be performed by using a modular or loosely coupled combination of existing results on ontology and rule updates. On the other hand, the general problem of hybrid updates is very complex as it requires the reconciliation and convergence of a number of fundamentally different approaches to knowledge dynamics. We have made the first steps in this respect, by identifying a unifying perspective that embraces model-based belief change operators as well as the historically first rule update semantics. Nevertheless, many exciting and challenging questions persist and need to be confronted by future research.

References

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2): 510–530, 1985. 9, 36, 42, 106

José Júlio Alferes and Luís Moniz Pereira. Update-programs can update programs. In Jürgen Dix, Luís Moniz Pereira, and Teodor C. Przymusinski, editors, *Non-Monotonic Extensions of Logic Programming (NMELP '96), Selected Papers*, volume 1216 of *Lecture Notes in Computer Science*, pages 110–131, Bad Honnef, Germany, September 5-6 1996. Springer. 11, 42, 44, 99

José Júlio Alferes, João Alexandre Leite, Luís Moniz Pereira, Halina Przymusinska, and Teodor C. Przymusinski. Dynamic updates of non-monotonic knowledge bases. *The Journal of Logic Programming*, 45(1-3):43–70, September/October 2000. 11, 27, 43, 45

José Júlio Alferes, Antonio Brogi, João Alexandre Leite, and Luís Moniz Pereira. An evolvable rule-based e-mail agent. In Fernando Moura-Pires and Salvador Abreu, editors, *Proceedings of the 11th Portuguese Conference Artificial Intelligence (EPIA 2003)*, volume 2902 of *Lecture Notes in Computer Science*, pages 394–408, Beja, Portugal, December 4-7 2003. Springer. 12

José Júlio Alferes, Federico Banti, Antonio Brogi, and João Alexandre Leite. The refined extension principle for semantics of dynamic logic programming. *Studia Logica*, 79(1): 7–32, 2005. 11, 27, 43, 46, 55, 131, 145

José Júlio Alferes, Matthias Knorr, and Terrance Swift. Queries to hybrid MKNF knowledge bases through oracular tabling. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, volume 5823 of *Lecture Notes in Computer Science*, pages 1–16, Chantilly, VA, USA, October 25-29 2009. Springer. 8

Krzysztof R. Apt and Marc Bezem. Acyclic programs. *New Generation Computing*, 9(3/4): 335–364, 1991. 25

Krzysztof R. Apt, Howard A. Blair, and Adrian Walker. Towards a theory of declarative knowledge. In *Foundations of Deductive Databases and Logic Programming*, pages 89–148. Morgan Kaufmann, 1988. 53

Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the el envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 364–369, Edinburgh, Scotland, UK, July 30-August 5 2005a. Professional Book Center. 41

Franz Baader, Carsten Lutz, Maja Milicic, Ulrike Sattler, and Frank Wolter. Integrating description logics and action formalisms: First results. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference (AAAI-05)*, pages 572–577, Pittsburgh, Pennsylvania, USA, July 9-13 2005b. AAAI Press / The MIT Press. 10

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2nd edition, 2007. 4, 5, 22, 23

Chitta Baral. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, 2003. 24

Sean Bechhofer, Ian Horrocks, Carole A. Goble, and Robert Stevens. OilEd: A Reasonable ontology editor for the semantic web. In Franz Baader, Gerhard Brewka, and Thomas Eiter, editors, *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (KI 2001)*, volume 2174 of *Lecture Notes in Computer Science*, pages 396–408, Vienna, Austria, September 19-21 2001. Springer. 9

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):28–37, 2001. 3

Yusri Bong. The description logic ABox update problem revisited. Master's thesis, Dresden University of Technology, Dresden, Germany, February 2007. 10, 36, 41

Craig Boutilier. Generalized update: Belief change in dynamic settings. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 1550–1556, Montréal, Québec, Canada, August 20-25 1995. Morgan Kaufmann. 34

Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation*. Morgan Kaufmann, Los Altos, California, August 1985. 153

Gerhard Brewka and Thomas Eiter. Equilibria in heterogeneous nonmonotonic multi-context systems. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 385–390, Vancouver, British Columbia, Canada, July 22-26 2007. AAAI Press. 73, 144

Gerhard Brewka and Joachim Hertzberg. How to do things with worlds: On formalizing actions and plans. *Journal of Logic and Computation*, 3(5):517–532, 1993. 34

Daniel R. Brooks, Esra Erdem, Selim T. Erdogan, James W. Minett, and Donald Ringe. Inferring phylogenetic trees using answer set programming. *Journal of Automated Reasoning*, 39(4):471–511, 2007. 6

Francesco Buccafurri, Wolfgang Faber, and Nicola Leone. Disjunctive logic programs with inheritance. In Danny De Schreye, editor, *Proceedings of the 1999 International Conference on Logic Programming (ICLP 1999)*, pages 79–93, Las Cruces, New Mexico, USA, November 29 - December 4 1999. The MIT Press. 43, 47

Pedro Cabalar and Paolo Ferraris. Propositional theories are strongly equivalent to logic programs. *Theory and Practice of Logic Programming (TPLP)*, 7(6):745–759, 2007. 102

Pedro Cabalar, David Pearce, and Agustín Valverde. Minimal logic programs. In Verónica Dahl and Ilkka Niemelä, editors, *Proceedings of the 23rd International Conference on Logic Programming (ICLP 2007)*, volume 4670 of *Lecture Notes in Computer Science*, pages 104–118, Porto, Portugal, September 8-13 2007. Springer. 111, 112, 113

Diego Calvanese, Evgeny Kharlamov, Werner Nutt, and Dmitriy Zheleznyakov. Evolution of DL-Lite knowledge bases. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *International Semantic Web Conference (1)*, volume 6496 of *Lecture Notes in Computer Science*, pages 112–128, Shanghai, China, November 7-11 2010. Springer. 13, 35, 41, 42, 90, 95, 137, 145

Jan Chomicki, Jorge Lobo, and Shamim A. Naqvi. A logic programming approach to conflict resolution in policy management. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 121–132, Breckenridge, Colorado, USA, April 11-15 2000. Morgan Kaufmann Publishers. 6

Mukesh Dalal. Investigations into a theory of knowledge base revision. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI 1988)*, pages 475–479, St. Paul, MN, USA, August 21-26 1988. AAAI Press / The MIT Press. 31, 66

Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artif. Intell.*, 89 (1-2):1–29, 1997. 145

Brian A. Davey and Hilary A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990. 115, 257

Randall Davis, Howard E. Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993. 3

Jos de Bruijn, David Pearce, Axel Polleres, and Agustín Valverde. A semantical framework for hybrid knowledge bases. *Journal of Knowledge and Information Systems*, 25(1): 81–104, 2010. 7, 8

Jos de Bruijn, Thomas Eiter, Axel Polleres, and Hans Tompits. Embedding nonground logic programs into autoepistemic logic for knowledge-base combination. *ACM Transactions on Computational Logic (TOCL)*, 12(3):20, 2011. 8

Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. On the update of description logic ontologies at the instance level. In *Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference*, Boston, Massachusetts, USA, July 16-20 2006. AAAI Press. 10, 37, 41

Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. On the approximation of instance level update and erasure in description logics. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*, pages 403–408, Vancouver, British Columbia, Canada, July 22-26 2007. AAAI Press. 10, 37, 41

Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. On instance-level update and erasure in description logic ontologies. *Journal of Logic and Computation*, 19(5):745–770, 2009. 10, 37, 41, 137

James P. Delgrande. A Program-Level Approach to Revising Logic Programs under the Answer Set Semantics. *Theory and Practice of Logic Programming, 26th Int'l. Conference on Logic Programming (ICLP'10) Special Issue*, 10(4-6):565–580, July 2010. 43, 52, 187

James P. Delgrande, Torsten Schaub, and Hans Tompits. A preference-based framework for updating logic programs. In Chitta Baral, Gerhard Brewka, and John S. Schlipf, editors, *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2007)*, volume 4483 of *Lecture Notes in Computer Science*, pages 71–83, Tempe, AZ, USA, May 15-17 2007. Springer. 12, 43, 49, 50, 51, 131, 187

James P. Delgrande, Torsten Schaub, Hans Tompits, and Stefan Woltran. Belief revision of logic programs under answer set semantics. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 411–421, Sydney, Australia, September 16-19 2008. AAAI Press. 99, 100, 101, 102, 104, 106

Jürgen Dix. A classification theory of semantics of normal logic programs: I. Strong properties. *Fundamenta Informaticae*, 22(3):227–255, 1995a. 67

Jürgen Dix. A classification theory of semantics of normal logic programs: II. Weak properties. *Fundamenta Informaticae*, 22(3):257–288, 1995b. 53

Patrick Doherty, Witold Lukaszewicz, and Ewa Madalinska-Bugaj. The PMA and relativizing minimal change for action update. In Anthony G. Cohn, Lenhard K. Schubert, and Stuart C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 258–269, Trento, Italy, June 2-5 1998. Morgan Kaufmann. 34, 35

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. AL-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems*, 10 (3):227–252, 1998. 6, 8

Conrad Drescher, Hongkai Liu, Franz Baader, Peter Steinke, and Michael Thielscher. Putting ABox updates into action. In *Proceedings of the 8th IJCAI International Workshop on Nonmontonic Reasoning, Action and Change (NRAC-09)*, 2009. 10, 36, 41

Thomas Eiter, Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. The diagnosis frontend of the dlv system. *AI Communications*, 12(1-2):99–111, 1999. 6

Thomas Eiter, Michael Fink, Giuliana Sabbatini, and Hans Tompits. On properties of update sequences based on causal rejection. *Theory and Practice of Logic Programming (TPLP)*, 2(6):721–777, 2002. 11, 14, 43, 44, 47, 68, 131, 145

Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits. Combining answer set programming with description logics for the semantic web. In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR2004)*, pages 141–151, Whistler, Canada, June 2-5 2004. AAAI Press. 8

Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 90–96, Edinburgh, Scotland, UK, July 30-August 5 2005. Professional Book Center. 8

Ronald Fagin, Jeffrey D. Ullman, and Moshe Y. Vardi. On the semantics of updates in databases. In *Proceedings of the 2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS 1986)*, pages 352–365, Colony Square Hotel, Atlanta, Georgia, March 21-23 1983. ACM. 35, 137

Melvin Fitting. *First-Order Logic and Automated Theorem Proving*. Graduate texts in computer science. Springer-Verlag, Berlin, Germany, 2nd edition, 1996. 21, 22

Giorgos Flouris. *On Belief Change and Ontology Evolution*. PhD thesis, University of Crete, 2006. 10

Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. Inconsistencies, negations and changes in ontologies. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, pages 1295–1300, Boston, Massachusetts, USA, July 16-20 2006a. AAAI Press. 10

Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. Evolving ontology evolution. In *Proceedings of the 32nd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 06)*, volume 3831 of *Lecture Notes in Computer Science*, pages 14–29, Merin, Czech Republic, January 21-27 2006b. Springer. Invited Talk. 10

Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. On generalizing the AGM postulates. In Loris Penserini, Pavlos Peppas, and Anna Perini, editors, *Proceedings of the 3rd Starting AI Researchers' Symposium (STAIRS 2006)*, volume 142 of *Frontiers in Artificial Intelligence and Applications*, pages 132–143, Riva del Garda, Trentino, Italy, 2006c. IOS Press. 10

Giorgos Flouris, Dimitris Makanatas, Haridimos Kondylakis, Dimitris Plexousakis, and Grigoris Antoniou. Ontology change: classification and survey. *The Knowledge Engineering Review*, 23(2):117–152, 2008. 9

Norman Y. Foo. Ontology revison. In Gerard Ellis, Robert Levinson, William Rich, and John F. Sowa, editors, *Proceedings of the 3rd International Conference on Conceptual Structures (ICCS '95)*, volume 954 of *Lecture Notes in Computer Science*, pages 16–31, Santa Cruz, California, USA, August 14-18 1995. Springer. 10

Kenneth D. Forbus. Introducing actions into qualitative simulation. In N. S. Sridharan, editor, *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1273–1278, Detroit, MI, USA, August 1989. Morgan Kaufmann. 33

André Fuhrmann. Theory contraction through base contraction. *Journal of Philosophical Logic*, 20:175–203, 1991. 94

Peter Gärdenfors. The dynamics of knowledge and belief: Foundational vs. coherence theories. *Revue Internationale de Philosophie*, 44:24–46, 1990. 94

Peter Gärdenfors. *Belief Revision*, chapter Belief Revision: An Introduction, pages 1–28. Cambridge University Press, 1992. 36

Martin Gebser, Benjamin Kaufmann, André Neumann, and Torsten Schaub. *Clasp*: A conflict-driven answer set solver. In Chitta Baral, Gerhard Brewka, and John S. Schlipf, editors, *Proceedings of the 9th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2007)*, volume 4483 of *Lecture Notes in Computer Science*, pages 260–265, Tempe, AZ, USA, May 15-17 2007. Springer. 6

Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991. 6, 146

Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert A. Kowalski and Kenneth A. Bowen, editors, *Proceedings of the 5th International Conference and Symposium on Logic Programming (ICLP/SLP 1988)*, pages 1070–1080, Seattle, Washington, August 15-19 1988. MIT Press. 6, 24, 62

Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3-4):365–385, 1991. 24, 25

Matthew L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30(1):35–79, 1986. 35

Matthew L. Ginsberg and David E. Smith. Reasoning about action i: A possible worlds approach. *Artificial Intelligence*, 35(2):165–195, 1988. 35

Ana Sofia Gomes, José Júlio Alferes, and Terrance Swift. Implementing query answering for hybrid MKNF knowledge bases. In Manuel Carro and Ricardo Peña, editors, *Proceedings of the 12th International Symposium on Practical Aspects of Declarative Languages (PADL 2010)*, volume 5937 of *Lecture Notes in Computer Science*, pages 25–39, Madrid, Spain, January 18-19 2010. Springer. 8

Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: Combining logic programs with description logic. In *Proceedings of the 12th International World Wide Web Conference (WWW 2003)*, pages 48–57, Budapest, Hungary, May 20-24 2003. ACM. 6, 8

Adam Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17(2): 157–170, 1988. 10

Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, June 1993. 4

Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman. The meaning of erasing in RDF under the Katsuno-Mendelzon approach. In *Proceedings of the 9th International Workshop on the Web and Databases (WebDB 2006)*, Chicago, Illinois, USA, June 30 2006. 13

Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman. RDFS update: From theory to practice. In Grigoris Antoniou, Marko Grobelnik, Elena Paslaru Bontas Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Z. Pan, editors, *Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011), Part II*, volume 6644 of *Lecture Notes in Computer Science*, pages 93–107, Heraklion, Crete, Greece, May 29 - June 2 2011. Springer. 13

Peter Haase and Ljiljana Stojanovic. Consistent evolution of OWL ontologies. In *Proceedings of the 2nd European Semantic Web Conference*, Heraklion, Greece, May 2005. 9

Peter Haase, Frank van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt, and York Sure. A framework for handling inconsistency in changing ontologies. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, volume 3729 of *Lecture Notes in Computer Science*, pages 353–367, Galway, Ireland, November 6-10 2005. Springer. 9

Christian Halaschek-Wiener and Yarden Katz. Belief base revision for expressive description logics. In Bernardo Cuenca Grau, Pascal Hitzler, Conor Shankey, and Evan Wallace, editors, *Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions*, volume 216 of *CEUR Workshop Proceedings*, Athens, Georgia, USA, November 10-11 2006. CEUR-WS.org. 10, 137

Sven Ove Hansson. Reversing the Levi identity. *Journal of Philosophical Logic*, 22(6):637–669, December 1993a. 36, 42, 136

Sven Ove Hansson. Theory contraction and base contraction unified. *Journal of Symbolic Logic*, 58(2):602–625, 1993b. 94

Sven Ove Hansson. Knowledge-level analysis of belief base operations. *Artificial Intelligence*, 82(1-2):215–235, 1996. 10

Patrick J. Hayes. The logic of frames. In Dieter Metzing, editor, *Frame conceptions and text understanding*, pages 46–61. Walter de Gruyter and Co., 1979. Republished in Brachman and Levesque (1985). 3

Andreas Herzig. The PMA revisited. In Luigia Carlucci Aiello, Jon Doyle, and Stuart C. Shapiro, editors, *Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 40–50, Cambridge, Massachusetts, USA, November 5-8 1996. Morgan Kaufmann. 34, 35, 41

Andreas Herzig. On updates with integrity constraints. In James P. Delgrande, Jérôme Lang, Hans Rott, and Jean-Marc Tallon, editors, *Belief Change in Rational Agents: Perspectives from Artificial Intelligence, Philosophy, and Economics*, volume 05321 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, August 7-12 2005. 35, 41

Andreas Herzig and Omar Rifi. Propositional belief base update and minimal change. *Artificial Intelligence*, 115(1):107–138, 1999. 10, 34, 91, 132, 289

Stijn Heymans, Livia Predoiu, Cristina Feier, Jos de Bruijn, and Davy Van Nieuwenborgh. G-hybrid knowledge bases. In *Proceedings of the ICLP'06 Workshop on Applications of Logic Programming in the Semantic Web and Semantic Web Services (ALPSWS 2006)*, 2006. 8

Arend Heyting. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften*, pages 42–56, 1930. Reprint in *Logik-Texte: Kommentierte Auswahl zur Geschichte der Modernen Logik*, Akademie-Verlag, 1986. 100, 102

Pascal Hitzler and Bijan Parsia. Ontologies and rules. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 111–132. Springer, Berlin, second edition, 2009. 7

Martin Homola. Dynamic logic programming: Various semantics are equal on acyclic programs. In João Alexandre Leite and Paolo Torroni, editors, *Proceedings of the 5th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA V)*, volume 3487 of *Lecture Notes in Computer Science*, pages 78–95, Lisbon, Portugal, September 29-30 2004. Springer. 46, 55, 187, 188

Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A semantic web rule language. W3C Member Submission 21 May 2004, http://www.w3.org/Submission/SWRL/, 2004. 8

Manoela Ilic, João Leite, and Martin Slota. Explicit dynamic user profiles for a collaborative filtering recommender system. In Hector Geffner, Rui Prada, Isabel Machado Alexandre, and Nuno David, editors, *Proceedings of the 11th Ibero-American Conference on Artificial Intelligence (IBERAMIA'08)*, volume LNAI 5290, pages 352–361. Springer-Verlag, 2008. 12

Katsumi Inoue and Chiaki Sakama. Negation as failure in the head. *Journal of Logic Programming*, 35(1):39–78, 1998. 111, 112

Katsumi Inoue and Chiaki Sakama. Equivalence of logic programs under updates. In José Júlio Alferes and João Alexandre Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, volume 3229 of *Lecture Notes in Computer Science*, pages 174–186, Lisbon, Portugal, September 27-30 2004. Springer. 111, 118

Tomi Janhunen. On the effect of default negation on the expressiveness of disjunctive rules. In Thomas Eiter, Wolfgang Faber, and Miroslaw Truszczynski, editors, *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2001)*, volume 2173 of *Lecture Notes in Computer Science*, pages 93–106, Vienna, Austria, September 17-19 2001. Springer. 111

Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In James F. Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 387–394, Cambridge, MA, USA, April 22-25 1991. Morgan Kaufmann Publishers. 9, 10, 30, 31, 33, 41, 99

Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1992. 106

Arthur M. Keller and Marianne Winslett. On the use of an extended relational model to handle changing incomplete information. *IEEE Transactions on Software Engineering*, 11 (7):620–633, 1985. 9, 10, 30

Michael Klein and Natalya F. Noy. A component-based framework for ontology evolution. In Fausto Giunchiglia, Asucion Gomez-Perez, Adam Pease, Heiner Stuckenschmidt, York Sure, and Steven Willmott, editors, *Proceedings of the IJCAI-03 Workshop on Ontologies and Distributed Systems, CEUR-WS*, volume 71 of *CEUR Workshop Proceedings*, Acapulco, Mexico, August 9 2003. CEUR-WS.org. 9

Matthias Knorr, José Júlio Alferes, and Pascal Hitzler:. Local closed world reasoning with description logics under the well-founded semantics. *Artificial Intelligence*, 175 (9-10):1528–1554, 2011. 8, 146

George Konstantinidis, Giorgos Flouris, Grigoris Antoniou, and Vassilis Christophides. On RDF/S ontology evolution. In Vassilis Christophides, Martine Collard, and Claudio Gutierrez, editors, *Proceedings of the VLDB Workshop on Semantic Web, Ontologies and Databases (SWDB-ODBIS 2007), Revised Selected Papers*, volume 5005 of *Lecture Notes in Computer Science*, pages 21–42, Vienna, Austria, September 24 2007. Springer. 13

George Konstantinidis, Giorgos Flouris, Grigoris Antoniou, and Vassilis Christophides. A formal approach for RDF/S ontology evolution. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikolaos M. Avouris, editors, *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 70–74, Patras, Greece, July 21-25 2008. IOS Press. 13

Robert A. Kowalski. Legislation as logic programs. In Gérard Comyn, Norbert E. Fuchs, and Michael Ratcliffe, editors, *Proceedings of the 2nd Second International Logic Programming Summer School (LPSS '92)*, volume 636 of *Lecture Notes in Computer Science*, pages 203–230, Zurich, Switzerland, September 7-11 1992. Springer. 6

Robert A. Kowalski. Workshop report: Application of logic programming to legal reasoning. *New Generation Computing*, 13(3&4):435–439, 1995. 6

Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Description logic rules. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikos Avouris, editors, *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI2008)*, pages 80–84, Patras, Greece, July 2008a. IOS Press. 8

Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. ELP: Tractable rules for OWL 2. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, editors, *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, volume 5318 of *Lecture Notes in Computer Science*, pages 649–664, Karlsruhe, Germany, October 26-30 2008b. Springer. 8

Patrick Krümpelmann. Dependency semantics for sequences of extended logic programs. *Logic Journal of the IGPL*, 2012. To appear. 12, 43, 53

Patrick Krümpelmann and Gabriele Kern-Isberner. On belief dynamics of dependency relations for extended logic programs. In *Proceedings of the 13th International Workshop on Non-Monotonic Reasoning*, Toronto, Canada, May 2010. 12, 43, 53

João Alexandre Leite. *Evolving Knowledge Bases*, volume 81 of *Frontiers of Artificial Intelligence and Applications, xviii + 307 p. Hardcover*. IOS Press, 2003. 43, 45, 46, 51, 67, 119, 128

João Alexandre Leite and Luís Moniz Pereira. Generalizing updates: From models to programs. In Jürgen Dix, Luís Moniz Pereira, and Teodor C. Przymusinski, editors, *Proceedings of the 3rd International Workshop on Logic Programming and Knowledge Representation (LPKR '97)*, volume 1471 of *Lecture Notes in Computer Science*, pages 224–246, Port Jefferson, New York, USA, October 17 1997. Springer. 11, 14, 27, 42, 43, 44, 99

João Alexandre Leite, José Júlio Alferes, and Belopeta Mito. Resource allocation with answer-set programming. In Carles Sierra, Cristiano Castelfranchi, Keith S. Decker, and Jaime Simão Sichman, editors, *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Budapest, Hungary, May 10-15 2009. IFAAMAS. 6

Maurizio Lenzerini and Domenico Fabio Savo. On the evolution of the instance level of DL-Lite knowledge bases. In Riccardo Rosati, Sebastian Rudolph, and Michael Zakharyaschev, editors, *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, volume 745 of *CEUR Workshop Proceedings*, Barcelona, Spain, July 13-16 2011. CEUR-WS.org. 13, 42, 136, 137

Nicola Leone, Gianluigi Greco, Giovambattista Ianni, Vincenzino Lio, Giorgio Terracina, Thomas Eiter, Wolfgang Faber, Michael Fink, Georg Gottlob, Riccardo Rosati, Domenico Lembo, Maurizio Lenzerini, Marco Ruzzi, Edyta Kalka, Bartosz Nowicki, and Witold Staniszkis. The INFOMIX system for advanced integration of incomplete and inconsistent data. In Fatma Özcan, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 915–917, Baltimore, Maryland, USA, June 14-16 2005. ACM. 6

Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7(3):499–562, 2006. 6

Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, 1984. 3

Alon Y. Levy and Marie-Christine Rousset. Combining horn rules and description logics in CARIN. *Artificial Intelligence*, 104(1-2):165–209, 1998. 7

Vladimir Lifschitz. Frames in the space of situations. *Artificial Intelligence*, 46(3):365–376, 1990. 35, 41

Vladimir Lifschitz. Nonmonotonic databases and epistemic queries. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 381–386, 1991. 26

Vladimir Lifschitz. Answer set planning. In Danny De Schreye, editor, *Logic Programming: The 1999 International Conference*, pages 23–37, Las Cruces, New Mexico, USA, November 29 - December 4 1999a. The MIT Press. 6

Vladimir Lifschitz. Action languages, answer sets and planning. In *The Logic Programming Paradigm: a 25-Year Perspective*, pages 357–373. Springer Verlag, 1999b. 6

Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In Pascal Van Hentenryck, editor, *Proceedings of the 11th International Conference on Logic Programming (ICLP 1994)*, pages 23–37, Santa Margherita Ligure, Italy, June 13-18 1994. MIT Press. 13, 15, 70, 72, 73, 75, 80, 212

Vladimir Lifschitz and Thomas Y. C. Woo. Answer sets in general nonmonotonic reasoning (preliminary report). In Bernhard Nebel, Charles Rich, and William R. Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 603–614, Cambridge, MA, October 25-29 1992. Morgan Kaufmann. 24

Vladimir Lifschitz, David Pearce, and Agustín Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL)*, 2(4):526–541, 2001. 67, 101, 118

Hongkai Liu, Carsten Lutz, Maja Miličić, and Frank Wolter. Updating description logic ABoxes. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 46–56, Lake District of the United Kingdom, June 2-5 2006. AAAI Press. 10, 36, 41, 137

Jan Łukasiewicz. Die Logik und das Grundlagenproblem. In *Les Entretiens de Zürich sue les Fondements et la méthode des sciences mathématiques 1938*, pages 82–100. Zürich, 1941. 102

David Makinson. General theory of cumulative inference. In Reinfrank Michael, Johan De Kleer, Matthew L. Ginsberg, and Erik Sandewall, editors, *Proceedings of the 2nd International Workshop on Non-Monotonic Reasoning (NMR'88)*, volume 346 of *Lecture Notes in Computer Science*, pages 1–18, Grassau, FRG, June 13-15 1988. Springer. 67

Viktor W. Marek and Miroslaw Truszczynski. Revision programming. *Theoretical Computer Science*, 190(2):241–277, 1998. 48

John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In Bernard Meltzer and Donald Michie, editors, *Machine Intelligence*, volume IV, pages 463–502. Edinburgh University Press, 1969. 3

Marvin Minsky. A framework for representing knowledge. *The Psychology of Computer Vision*, page 211–277, 1975. 3

Boris Motik and Riccardo Rosati. A faithful integration of description logics with logic programming. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 477–482, Hyderabad, India, January 6-12 2007. 21, 26

Boris Motik and Riccardo Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5):93–154, 2010. 8, 12, 21, 26, 27, 28, 29, 30, 61

Boris Motik, Ian Horrocks, Riccardo Rosati, and Ulrike Sattler. Can OWL and logic programming live together happily ever after? In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, volume 4273 of *LNCS*, pages 501–514, Athens, GA, USA, Nov 5-9 2006. Springer. 6

Nadeschda Nikitina, Sebastian Rudolph, and Birte Glimm. Reasoning-supported interactive revision of knowledge bases. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1027–1032, Barcelona, Catalonia, Spain, July 16-22 2011. IJCAI/AAAI. 9

Monica Nogueira, Marcello Balduccini, Michael Gelfond, Richard Watson, and Matthew Barry. An A-Prolog decision support system for the space shuttle. In *Proceedings of the 1st International Workshop on Answer Set Programming (ASP'01)*, pages 169–183, Stanford, March 26-28 2001. 6

Natalya F. Noy and Michel C. A. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428–440, 2004. 9

Natalya F. Noy, Abhita Chugh, William Liu, and Mark A. Musen. A framework for ontology evolution in collaborative environments. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, 2006. 9

Natalya Fridman Noy, Ray W. Fergerson, and Mark A. Musen. The knowledge model of Protégé-2000: Combining interoperability and flexibility. In Rose Dieng and Olivier Corby, editors, *Proceedings of the 12th International Conference on Knowledge Acquisition, Modeling and Management (EKAW 2000)*, volume 1937 of *Lecture Notes in Computer Science*, pages 17–32, Juan-les-Pins, France, October 2-6 2000. Springer. 9

Mauricio Osorio and Víctor Cuevas. Updates in answer set programming: An approach based on basic structural properties. *Theory and Practice of Logic Programming*, 7(4): 451–479, 2007. 11, 43, 47, 50

Mauricio Osorio and Claudia Zepeda. Update sequences based on minimal generalized pstable models. In Alexander F. Gelbukh and Angel Fernando Kuri Morales, editors, *Proceedings of the 6th Mexican International Conference on Artificial Intelligence (MICAI 2007)*, volume 4827 of *Lecture Notes in Computer Science*, pages 283–293, Aguascalientes, Mexico, November 4-10 2007. Springer. 43, 53

David Pearce. A new logical characterisation of stable models and answer sets. In Jürgen Dix, Luís Moniz Pereira, and Teodor C. Przymusinski, editors, *Proceedings of the 6th Workshop on Non-Monotonic Extensions of Logic Programming (NMELP '96)*, volume 1216 of *Lecture Notes in Computer Science*, pages 57–70, Bad Honnef, Germany, September 5-6 1997. Springer. 100, 102

Peter Plessers and Olga De Troyer. Ontology change detection using a version log. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, volume 3729 of *Lecture Notes in Computer Science*, pages 578–592, Galway, Ireland, November 6-10 2005. Springer. 9

Henry Prakken and Giovanni Sartor. The role of logic in computational models of legal argument: A critical survey. In Antonis C. Kakas and Fariba Sadri, editors, *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II*, volume 2408 of *Lecture Notes in Computer Science*, pages 342–381. Springer, 2002. 6

Guilin Qi and Jianfeng Du. Model-based revision operators for terminologies in description logics. In Craig Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena (IJCAI 2009)*, pages 891–897, California, USA, July 11-17 2009. 10

Guilin Qi and Fangkai Yang. A survey of revision approaches in description logics. In Diego Calvanese and Georg Lausen, editors, *Proceedings of the 2nd International Conference on Web Reasoning and Rule Systems (RR'08)*, volume 5341 of *Lecture Notes in Computer Science*, pages 74–88, Karlsruhe, Germany, October 31-November 1 2008. Springer Verlag. 10, 137

Guilin Qi, Weiru Liu, and David A. Bell. Knowledge base revision in description logics. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *Proceedings of the 10th European Conference on the Logics in Artificial Intelligence (JELIA'06)*, volume 4160 of *Lecture Notes in Computer Science*, pages 386–398, Liverpool, UK, September 13-15 2006a. Springer. 10

Guilin Qi, Weiru Liu, and David A. Bell. A revision-based approach to handling inconsistency in description logics. In *Proceedings of the 11th International workshop on Non-Monotonic Reasoning (NMR'06)*, 2006b. 10

Guilin Qi, Weiru Liu, and David A. Bell. A revision-based approach to handling inconsistency in description logics. *Journal of Artificial Intelligence Review*, 26(1-2):115–128, 2006c. 10

Guilin Qi, Peter Haase, Zhisheng Huang, Qiu Ji, Jeff Z. Pan, and Johanna Völker. A kernel revision operator for terminologies – algorithms and evaluation. In *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, volume 5318 of *Lecture Notes in Computer Science*, pages 419–434, Karlsruhe, Germany, October 26-30 2008. Springer. 10, 137

Prasad Rao, Konstantinos F. Sagonas, Terrance Swift, David Scott Warren, and Juliana Freire. XSB: A system for effciently computing WFS. In Jürgen Dix, Ulrich Furbach, and Anil Nerode, editors, *Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'97)*, volume 1265 of *Lecture Notes in Computer Science*, pages 431–441, Dagstuhl Castle, Germany, July 28-31 1997. Springer. 6

Raymond Reiter. What should a database know? *Journal of Logic Programming*, 14(1-2): 127–153, 1992. 7

Márcio Moretto Ribeiro and Renata Wassermann. Base revision in description logics - preliminary results. In *Proceedings of the International Workshop on Ontology Dynamics (IWOD-07)*, pages 69–82, 2007. 10, 137

Riccardo Rosati. Towards expressive KR systems integrating datalog and description logics: preliminary report. In Patrick Lambrix, Alexander Borgida, Maurizio Lenzerini, Ralf Möller, and Peter F. Patel-Schneider, editors, *Proceedings of the 1999 International Workshop on Description Logics (DL'99)*, volume 22 of *CEUR Workshop Proceedings*, Linköping, Sweden, July 30 - August 1 1999. CEUR-WS.org. 8

Riccardo Rosati. On the decidability and complexity of integrating ontologies and rules. *Journal of Web Semantics*, 3(1):61–73, 2005a. 8

Riccardo Rosati. Semantic and computational advantages of the safe integration of ontologies and rules. In François Fages and Sylvain Soliman, editors, *Proceedings of the 3rd International Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR 2005)*, volume 3703 of *Lecture Notes in Computer Science*, pages 50–64, Dagstuhl Castle, Germany, September 11-16 2005b. Springer. 8

Riccardo Rosati. DL+log: Tight integration of description logics and disjunctive datalog. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR2006)*, pages 68–78, Lake District of the United Kingdom, June 2-5 2006. AAAI Press. 8

Hans Rott. Two methods of constructing contractions and revisions of knowledge systems. *Journal of Philosophical Logic*, 20(2):149–173, 1991. 10

José Saias and Paulo Quaresma. A methodology to create legal ontologies in a logic programming based web information retrieval system. *Artificial Intelligence and Law*, 12(4):397–417, 2004. 12

Chiaki Sakama and Katsumi Inoue. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming (TPLP)*, 3(6):671–713, 2003. 12, 43, 51

Torsten Schaub and Kewen Wang. A semantic framework for preference handling in answer set programming. *Theory and Practice of Logic Programming (TPLP)*, 3(4-5):569–607, 2003. 49, 50, 187

Ján Šefránek. Irrelevant updates and nonmonotonic assumptions. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *Proceedings of the 10th European Conference on Logics in Artificial Intelligence (JELIA 2006)*, volume 4160 of *Lecture Notes in Computer Science*, pages 426–438, Liverpool, UK, September 13-15 2006. Springer. 12, 53, 145

Ján Šefránek. Static and dynamic semantics: Preliminary report. *Mexican International Conference on Artificial Intelligence*, pages 36–42, 2011. 12, 43, 53, 145

Marek J. Sergot, Fariba Sadri, Robert A. Kowalski, F. Kriwaczek, Peter Hammond, and H. T. Cory. The British Nationality Act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986. 6

Jozef Siska. Dynamic logic programming and world state evaluation in computer games. In Michael Fink, Hans Tompits, and Stefan Woltran, editors, *Proceedings of the 20th Workshop on Logic Programming*, volume 1843-06-02 of *INFSYS Research Report*, pages 64–70, Vienna, Austria, February 22-24 2006. Technische Universität Wien, Austria. 12

Martin Slota and João Leite. Marrying stable models with belief update. In *Proceedings of the 13th International Workshop on Non-Monotonic Reasoning (NMR 2010)*, 2010a. 17, 60

Martin Slota and João Leite. Towards Closed World Reasoning in Dynamic Open Worlds. *Theory and Practice of Logic Programming, 26th Int'l. Conference on Logic Programming (ICLP'10) Special Issue*, 10(4-6):547–564, July 2010b. 13, 17, 60

Martin Slota and João Leite. On semantic update operators for answer-set programs. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 957–962, Lisbon, Portugal, August 16-20 2010c. IOS Press. 17, 100

Martin Slota and João Leite. Back and forth between rules and SE-models. In James P. Delgrande and Wolfgang Faber, editors, *Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-11)*, volume 6645 of *Lecture Notes in Computer Science*, pages 174–186, Vancouver, Canada, May 16-19 2011. Springer. 17, 110

Martin Slota and João Leite. Robust equivalence models for semantic updates of answer-set programs. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning*, pages 158–168, Rome, Italy, June 10-14 2012a. AAAI Press. 17, 110, 124

Martin Slota and João Leite. A unifying perspective on knowledge updates. In Luis Fariñas del Cerro, Andreas Herzig, and Jérôme Mengin, editors, *Proceedings of the 13th European Conference on Logics in Artificial Intelligence (JELIA 2012)*, volume 7519 of *Logics in Artificial Intelligence (LNAI)*, pages 372–384, Toulouse, France, September 26-28 2012b. Springer. 17, 124

Martin Slota, João Leite, and Terrance Swift. Splitting and updating hybrid knowledge bases. *Theory and Practice of Logic Programming, 27th Int'l. Conference on Logic Programming (ICLP'11) Special Issue*, 11(4-5):801–819, 2011. 17, 72

Timo Soininen, Ilkka Niemelä, Juha Tiihonen, and Reijo Sulonen. Representing configuration knowledge with weight constraint rules. In Alessandro Provetti, Tran Cao Son, Alessandro Provetti, and Tran Cao Son, editors, *Answer Set Programming*, Stanford, March 26-28 2001. 6

Tran Cao Son and Jorge Lobo. Reasoning about policies using logic programs. In Alessandro Provetti and Tran Cao Son, editors, *Proceedings of the 1st International Workshop on Answer Set Programming (ASP'01)*, Stanford, March 26-28 2001. 6

Ljiljana Stojanovic and Boris Motik. Ontology evolution within ontology editors. In *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, Siguenza, Spain, September 30 2002. 9

Ljiljana Stojanovic, Alexander Maedche, Boris Motik, and Nenad Stojanovic. User-driven ontology evolution management. In Asunción Gómez-Pérez and V. Richard Benjamins, editors, *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, volume 2473 of *Lecture Notes in Computer Science*, pages 285–300, Siguenza, Spain, October 1-4 2002. Springer. 9

Ljiljana Stojanovic, Alexander Maedche, Nenad Stojanovic, and Rudi Studer. Ontology evolution as reconfiguration-design problem solving. In John H. Gennari, Bruce W. Porter, and Yolanda Gil, editors, *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003)*, pages 162–171, Sanibel Island, FL, USA, October 23-25 2003. ACM. 9

Heiner Stuckenschmidt and Michel C. A. Klein. Integrity and change in modular ontologies. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 900–908, Acapulco, Mexico, August 9-15 2003. Morgan Kaufmann. 9

V. S. Subrahmanian and Carlo Zaniolo. Relating stable models and AI planning domains. In Leon Sterling, editor, *Proceedings of the 12th International Conference on Logic Programming (ICLP 1995)*, pages 233–247, Tokyo, Japan, June 13-16 1995. MIT Press. 6

York Sure, Jürgen Angele, and Steffen Staab. Ontoedit: Multifaceted inferencing for ontology engineering. *Journal on Data Semantics I*, 2800:128–152, 2003. 9

Tommi Syrjänen and Ilkka Niemelä. The Smodels system. In Thomas Eiter, Wolfgang Faber, and Miroslaw Truszczynski, editors, *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2001)*, volume 2173 of *Lecture Notes in Computer Science*, pages 434–438, Vienna, Austria, September 17-19 2001. Springer. 6

Juha Tiihonen, Timo Soininen, Ilkka Niemelä, and Reijo Sulonen. A practical tool for mass-customising configurable products. In *Proceedings of the 14th International Conference on Engineering Design (ICED 03)*, pages 1290–1299, 2003. 6

Hudson Turner. Splitting a default theory. In *Proceedings of the 13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference (AAAI 96, IAAI 96)*, volume 1, pages 645–651, Portland, Oregon, August 4-8 1996. AAAI Press / The MIT Press. 73, 201

Hudson Turner. Strong equivalence made easy: nested expressions and weight constraints. *Theory and Practice of Logic Programming (TPLP)*, 3(4-5):609–622, 2003. 99, 100, 101

Frank Veltman. Making counterfactual assumptions. *Journal of Semantics*, 22(2):159–180, 2005. 35

Zhe Wang, Kewen Wang, and Rodney W. Topor. Revision of DL-Lite knowledge bases. In Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors, *Proceedings of the 22nd International Workshop on Description Logics (DL 2009)*, volume 477 of *CEUR Workshop Proceedings*, Oxford, UK, July 27-30 2009. CEUR-WS.org. 10

Zhe Wang, Kewen Wang, and Rodney W. Topor. A new approach to knowledge base revision in DL-Lite. In Maria Fox and David Poole, editors, *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*, Atlanta, Georgia, USA, July 11-15 2010. AAAI Press. 10

Renata Wassermann. Revising concepts. In *Proceedings of the 5th Workshop on Logic, Language, Information and Communication (WoLLIC-98)*, São Paulo, Brasil, 1998. 10

Renata Wassermann and Eduardo L. Fermé. A note on prototype revision. In *Spinning Idea (Electronic Essays dedicated to Peter Gärdenfors on his 50th Birthday), also available as Research Report PP-1999-18*. ILLC Series, 1999. 10

Marianne Winslett. Reasoning about action using a possible models approach. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI 1988)*, pages 89–93, Saint Paul, MN, USA, August 21-26 1988. AAAI Press / The MIT Press. 33

Marianne Winslett. *Updating Logical Databases*. Cambridge University Press, New York, USA, 1990. 10, 34, 35, 36, 38, 100, 104, 137

Stefan Woltran. A common view on strong, uniform, and other notions of equivalence in answer-set programming. *TPLP*, 8(2):217–234, 2008. 118

Ka-Shu Wong. A stronger notion of equivalence for logic programs. In Véronica Dahl and Ilkka Niemelä, editors, *Proceedings of the 23rd International Conference on Logic Programming (ICLP 2007)*, volume 4670 of *Lecture Notes in Computer Science*, pages 453–454, Porto, Portugal, September 8-13 2007. Springer. 117

Fangkai Yang, Guilin Qi, and Zhisheng Huang. A distance-based operator to revising ontologies in DL $\mathcal{SHOQ}$. In Claudio Sossai and Gaetano Chemello, editors, *Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009)*, volume 5590 of *Lecture Notes in Computer Science*, pages 434–445, Verona, Italy, July 1-3 2009. Springer. 10

Yan Zhang. Two results for prioritized logic programming. *Theory and Practice of Logic Programming (TPLP)*, 3(2):223–242, 2003. 48, 49

Yan Zhang. Logic program-based updates. *ACM Transactions on Computational Logic*, 7 (3):421–472, 2006. 12, 43, 48, 49, 50

Yan Zhang and Norman Y. Foo. Updating knowledge bases with disjunctive information. In *Proceedings of the 13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference, (AAAI/IAAI 96)*, volume 1, pages 562–568, Portland, Oregon, August 4-8 1996. AAAI Press / The MIT Press. 34

Yan Zhang and Norman Y. Foo. Updating logic programs. In Henri Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI 1998)*, pages 403–407, Brighton, UK, August 23-28 1998. John Wiley and Sons, Chichester. 49

Yan Zhang and Norman Y. Foo. Updates with disjunctive information: From syntactical and semantical perspectives. *Computational Intelligence*, 16(1):29–52, 2000. 34

Yan Zhang and Norman Y. Foo. A unified framework for representing logic program updates. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 707–713, Pittsburgh, Pennsylvania, USA, July 9-13 2005. AAAI Press / The MIT Press. 12, 47

Dmitriy Zheleznyakov, Diego Calvanese, Evgeny Kharlamov, and Werner Nutt. Updating TBoxes in DL-Lite. In Volker Haarslev, David Toman, and Grant E. Weddell, editors, *Proceedings of the 23rd International Workshop on Description Logics (DL 2010)*, volume 573 of *CEUR Workshop Proceedings*, Waterloo, Ontario, Canada, May 4-7 2010. CEUR-WS.org. 13

Part V

Proofs

A

Proofs: Background

It is at this point that normal language gives up, and goes and has a drink.

Terry Pratchett
The Color of Magic

This appendix contains proofs of propositions and theorems found in Chapter 2.

In addition, in Section A.2 we define and examine the properties of some additional concepts that are useful for manipulating MKNF knowledge bases in Chapters 2, 3 and 4. The statements and proofs in Sections A.2.4, A.2.5 and A.2.6 assume that the use of the equality predicate $\approx$ is not allowed.

A.1 First-Order Logic

Proposition 2.5. *Let $\mathcal{T}$, $\mathcal{S}$ be first-order theories. Then $\mathcal{T} \models_{\mathsf{FO}} \mathcal{S}$ implies $\mathcal{T} \models \mathcal{S}$ but the converse implication does not in general hold.*

Proof. Take some $I \in \mathcal{I}$ such that $I \models \mathcal{T}$, we need to show that $I \models \mathcal{S}$. For every constant symbol a we will denote the equivalence class induced by $\approx^I$ to which a belongs by $[a]$, i.e.

$$[a] = \left\{ b \in \mathcal{C} \;\middle|\; a \approx^I b \right\} \quad .$$

Now define a first-order interpretation I' over the universe $\Delta = \{ [a] \mid a \in \mathcal{C} \}$ as follows:

- for every $a \in \mathcal{C}$, $a^{I'} = [a]$;
- for every predicate symbol $P \in \mathcal{P}$ of arity n and all $d_1, d_2, \ldots, d_n \in \Delta$ such that

$d_i = [a_i]$ for some $a_i \in \mathcal{C}$,

$$(d_1, d_2, \dots, d_n) \in P^{I'} \qquad \text{if and only if} \qquad (a_1, a_2, \dots, a_n) \in P^I \ .$$

Note that this definition is unambiguous because I allows for replacement of equals by equals. It can be straightforwardly verified by induction on the structure of first-order sentences that for every first-order theory $\mathcal{T}'$,

$$I \models \mathcal{T}' \qquad \text{if and only if} \qquad I' \models \mathcal{T}' \ . \tag{A.1}$$

Thus, it follows from $I \models \mathcal{T}$ that $I' \models \mathcal{T}$. By our assumption that $\mathcal{T} \models_{\mathsf{FO}} \mathcal{S}$ we now obtain that $I' \models \mathcal{S}$ and using (A.1) we can conclude that $I \models \mathcal{S}$, which is the desired result.

To see that the converse implication does not hold, consider the theories

$$\mathcal{T} = \{ P(a) \mid a \in \mathcal{C} \} \qquad \text{and} \qquad \mathcal{S} = \{ \forall \mathbf{x} : P(\mathbf{x}) \} \ .$$

It is not difficult to verify that $\mathcal{T} \equiv \mathcal{S}$. We will construct a first-order interpretation I that satisfies $\mathcal{T}$ though it does not satisfy $\mathcal{S}$. Put $\Delta = \mathcal{C} \cup \{ d \}$ where $d \notin \mathcal{C}$ is a fresh object, $a^I = a$ for all $a \in \mathcal{C}$ and $P^I = \mathcal{C}$. Clearly, $I \models \mathcal{T}$ although $I \not\models \mathcal{S}$. □

A.2 MKNF Knowledge Bases

A.2.1 First-Order Sentences in MKNF Structures

Proposition A.1. *Let ϕ be a first-order sentence and $\mathcal{M}, \mathcal{N}$ be MKNF interpretations such that $\mathcal{M} \subseteq \mathcal{N}$. Then*

$$\mathcal{N} \models \mathbf{K}\,\phi \qquad \textit{implies} \qquad \mathcal{M} \models \mathbf{K}\,\phi \ .$$

Proof. This follows by the definition of MKNF entailment as follows:

$$\mathcal{N} \models \mathbf{K}\,\phi \Longrightarrow \forall I \in \mathcal{N} : (I, \mathcal{N}, \mathcal{N}) \models \phi \Longrightarrow \forall I \in \mathcal{N} : I \models \phi$$
$$\Longrightarrow \forall I \in \mathcal{M} : I \models \phi \Longrightarrow \forall I \in \mathcal{M} : (I, \mathcal{M}, \mathcal{M}) \models \phi \Longrightarrow \mathcal{M} \models \mathbf{K}\,\phi \ . \qquad \square$$

Proposition A.2. *Let ϕ be a first-order sentence, $\mathcal{S}$ a set of MKNF interpretations and $\mathcal{M} = \bigcup \mathcal{S}$. Then*

$$\forall \mathcal{N} \in \mathcal{S} : \mathcal{N} \models \mathbf{K}\,\phi \qquad \textit{implies} \qquad \mathcal{M} \models \mathbf{K}\,\phi \ .$$

Proof. Suppose that $\mathcal{N} \models \mathbf{K}\,\phi$ for all $\mathcal{N} \in \mathcal{S}$. In order to prove that $\mathcal{M} \models \mathbf{K}\,\phi$, take some $I \in \mathcal{M}$ and some $\mathcal{N} \in \mathcal{S}$ such that $I \in \mathcal{N}$. Since $\mathcal{N} \models \mathbf{K}\,\phi$, we obtain $(I, \mathcal{N}, \mathcal{N}) \models \phi$ and it follows that $I \models \phi$ because ϕ is a first-order sentence. Consequently, $(I, \mathcal{M}, \mathcal{M}) \models \phi$. As I was chosen arbitrarily, this holds for all $I \in \mathcal{M}$. Consequently, $\mathcal{M} \models \mathbf{K}\,\phi$. □

A.2.2 Subjective MKNF Sentences and Theories

Definition A.3 (Subjective, **K**-free and **not**-free MKNF Formula and Theory)**.** An MKNF formula ϕ is *subjective* if all first-order atoms in ϕ occur within the scope of a modal operator; **K**-*free* if it does not contain any occurrence of **K**; **not**-*free* if it does not contain any occurrence of **not**.

An MKNF theory is *subjective*, **K**-*free* and **not**-*free* if all its members are subjective, **K**-free and **not**-free, respectively.

Lemma A.4. *Let ϕ be a subjective MKNF sentence and $\mathcal{T}$ a subjective MKNF theory. For all $I_1, I_2 \in \mathfrak{I}$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ it holds that*

$$\begin{array}{lll} (I_1, \mathcal{M}, \mathcal{N}) \models \phi & \textit{if and only if} & (I_2, \mathcal{M}, \mathcal{N}) \models \phi \ , \\ (I_1, \mathcal{M}, \mathcal{N}) \models \mathcal{T} & \textit{if and only if} & (I_2, \mathcal{M}, \mathcal{N}) \models \mathcal{T} \ . \end{array}$$

Proof. We prove the first claim by induction on the structure of ϕ:

1° If $\phi = \mathbf{K}\,\psi$ for some MKNF sentence ψ, then it follows that

$$(I_1, \mathcal{M}, \mathcal{N}) \models \phi \iff \forall J \in \mathcal{M} : (J, \mathcal{M}, \mathcal{N}) \models \psi \iff (I_2, \mathcal{M}, \mathcal{N}) \models \phi \ .$$

2° If $\phi = \mathbf{not}\,\psi$ for some MKNF sentence ψ, then it follows that

$$(I_1, \mathcal{M}, \mathcal{N}) \models \phi \iff \exists J \in \mathcal{N} : (J, \mathcal{M}, \mathcal{N}) \not\models \psi \iff (I_2, \mathcal{M}, \mathcal{N}) \models \phi \ .$$

3° If $\phi = \neg\psi$, then using the inductive hypothesis for ψ we obtain

$$(I_1, \mathcal{M}, \mathcal{N}) \models \phi \iff (I_1, \mathcal{M}, \mathcal{N}) \not\models \psi \iff (I_2, \mathcal{M}, \mathcal{N}) \not\models \psi \iff (I_2, \mathcal{M}, \mathcal{N}) \models \phi \ .$$

4° If $\phi = \psi_1 \wedge \psi_2$, then using the inductive hypothesis for ψ_1 and ψ_2 we obtain

$$\begin{aligned} (I_1, \mathcal{M}, \mathcal{N}) \models \phi &\iff (I_1, \mathcal{M}, \mathcal{N}) \models \psi_1 \wedge (I_1, \mathcal{M}, \mathcal{N}) \models \psi_2 \\ &\iff (I_2, \mathcal{M}, \mathcal{N}) \models \psi_1 \wedge (I_2, \mathcal{M}, \mathcal{N}) \models \psi_2 \iff (I_2, \mathcal{M}, \mathcal{N}) \models \phi \ . \end{aligned}$$

5° If $\phi = \exists \mathbf{x} : \psi$, then using the inductive hypothesis for $\psi[a/\mathbf{x}]$ we obtain

$$\begin{aligned} (I_1, \mathcal{M}, \mathcal{N}) \models \phi &\iff \exists a \in \mathcal{C} : (I_1, \mathcal{M}, \mathcal{N}) \models \psi[a/\mathbf{x}] \\ &\iff \exists a \in \mathcal{C} : (I_2, \mathcal{M}, \mathcal{N}) \models \psi[a/\mathbf{x}] \iff (I_2, \mathcal{M}, \mathcal{N}) \models \phi \ . \end{aligned}$$

The second claim can be proved using the first one as follows:

$$\begin{aligned} (I_1, \mathcal{M}, \mathcal{N}) \models \mathcal{T} &\iff \forall \phi \in \mathcal{T} : (I_1, \mathcal{M}, \mathcal{N}) \models \phi \\ &\iff \forall \phi \in \mathcal{T} : (I_2, \mathcal{M}, \mathcal{N}) \models \phi \iff (I_2, \mathcal{M}, \mathcal{N}) \models \mathcal{T} \ . \end{aligned}$$ □

Lemma A.5. *Let ϕ be a* **not***-free MKNF sentence and $\mathcal{T}$ a* **not***-free MKNF theory. For all $I \in \mathfrak{I}$ and $\mathcal{M}, \mathcal{N}_1, \mathcal{N}_2 \in \mathfrak{M}$ it holds that*

$$\begin{array}{lll} (I, \mathcal{M}, \mathcal{N}_1) \models \phi & \textit{if and only if} & (I, \mathcal{M}, \mathcal{N}_2) \models \phi \ , \\ (I, \mathcal{M}, \mathcal{N}_1) \models \mathcal{T} & \textit{if and only if} & (I, \mathcal{M}, \mathcal{N}_2) \models \mathcal{T} \ . \end{array}$$

Proof. We prove the first claim by induction on the structure of ϕ:

1° If $\phi = p$ for some ground first-order atom p, then it follows that

$$(I, \mathcal{M}, \mathcal{N}_1) \models \phi \iff I \models p \iff (I, \mathcal{M}, \mathcal{N}_2) \models \phi \ .$$

2° If $\phi = \neg\psi$, then using the inductive hypothesis for ψ we obtain

$$(I, \mathcal{M}, \mathcal{N}_1) \models \phi \iff (I, \mathcal{M}, \mathcal{N}_1) \not\models \psi \iff (I, \mathcal{M}, \mathcal{N}_2) \not\models \psi \iff (I, \mathcal{M}, \mathcal{N}_2) \models \phi \ .$$

3° If $\phi = \psi_1 \wedge \psi_2$, then using the inductive hypothesis for ψ_1 and ψ_2 we obtain

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}_1) \models \phi &\iff (I, \mathcal{M}, \mathcal{N}_1) \models \psi_1 \wedge (I, \mathcal{M}, \mathcal{N}_1) \models \psi_2 \\ &\iff (I, \mathcal{M}, \mathcal{N}_2) \models \psi_1 \wedge (I, \mathcal{M}, \mathcal{N}_2) \models \psi_2 \iff (I, \mathcal{M}, \mathcal{N}_2) \models \phi \ .\end{aligned}$$

4° If $\phi = \exists \mathbf{x} : \psi$, then using the inductive hypothesis for $\psi[a/\mathbf{x}]$ we obtain

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}_1) \models \phi &\iff \exists a \in \mathcal{C} : (I, \mathcal{M}, \mathcal{N}_1) \models \psi[a/\mathbf{x}] \\ &\iff \exists a \in \mathcal{C} : (I, \mathcal{M}, \mathcal{N}_2) \models \psi[a/\mathbf{x}] \iff (I, \mathcal{M}, \mathcal{N}_2) \models \phi \ .\end{aligned}$$

5° If $\phi = \mathbf{K}\,\psi$, then using the inductive hypothesis for ψ we obtain

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}_1) \models \phi &\iff \forall J \in \mathcal{M} : (J, \mathcal{M}, \mathcal{N}_1) \models \psi \\ &\iff \forall J \in \mathcal{M} : (J, \mathcal{M}, \mathcal{N}_2) \models \psi \iff (I, \mathcal{M}, \mathcal{N}_2) \models \phi \ .\end{aligned}$$

The second claim can be proved using the first one as follows:

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}_1) \models \mathcal{T} &\iff \forall \phi \in \mathcal{T} : (I, \mathcal{M}, \mathcal{N}_1) \models \phi \\ &\iff \forall \phi \in \mathcal{T} : (I, \mathcal{M}, \mathcal{N}_2) \models \phi \iff (I, \mathcal{M}, \mathcal{N}_2) \models \mathcal{T} \ .\end{aligned}$$ □

Lemma A.6. *Let ϕ be a **K**-free MKNF sentence and $\mathcal{T}$ a **K**-free MKNF theory. For all $I \in \mathfrak{I}$ and $\mathcal{M}_1, \mathcal{M}_2, \mathcal{N} \in \mathfrak{M}$ it holds that*

$$\begin{aligned}(I, \mathcal{M}_1, \mathcal{N}) \models \phi \quad &\text{if and only if} \quad (I, \mathcal{M}_2, \mathcal{N}) \models \phi \ , \\ (I, \mathcal{M}_1, \mathcal{N}) \models \mathcal{T} \quad &\text{if and only if} \quad (I, \mathcal{M}_2, \mathcal{N}) \models \mathcal{T} \ .\end{aligned}$$

Proof. We prove the first claim by induction on the structure of ϕ:

1° If $\phi = p$ for some ground first-order atom p, then it follows that

$$(I, \mathcal{M}_1, \mathcal{N}) \models \phi \iff I \models p \iff (I, \mathcal{M}_2, \mathcal{N}) \models \phi \ .$$

2° If $\phi = \neg\psi$, then using the inductive hypothesis for ψ we obtain

$$(I, \mathcal{M}_1, \mathcal{N}) \models \phi \iff (I, \mathcal{M}_1, \mathcal{N}) \not\models \psi \iff (I, \mathcal{M}_2, \mathcal{N}) \not\models \psi \iff (I, \mathcal{M}_2, \mathcal{N}) \models \phi \ .$$

3° If $\phi = \psi_1 \wedge \psi_2$, then using the inductive hypothesis for ψ_1 and ψ_2 we obtain

$$\begin{aligned}(I, \mathcal{M}_1, \mathcal{N}) \models \phi &\iff (I, \mathcal{M}_1, \mathcal{N}) \models \psi_1 \wedge (I, \mathcal{M}_1, \mathcal{N}) \models \psi_2 \\ &\iff (I, \mathcal{M}_2, \mathcal{N}) \models \psi_1 \wedge (I, \mathcal{M}_2, \mathcal{N}) \models \psi_2 \iff (I, \mathcal{M}_2, \mathcal{N}) \models \phi \ .\end{aligned}$$

4° If $\phi = \exists \mathbf{x} : \psi$, then using the inductive hypothesis for $\psi[a/\mathbf{x}]$ we obtain

$$\begin{aligned}(I, \mathcal{M}_1, \mathcal{N}) \models \phi &\iff \exists a \in \mathcal{C} : (I, \mathcal{M}_1, \mathcal{N}) \models \psi[a/\mathbf{x}] \\ &\iff \exists a \in \mathcal{C} : (I, \mathcal{M}_2, \mathcal{N}) \models \psi[a/\mathbf{x}] \iff (I, \mathcal{M}_2, \mathcal{N}) \models \phi \ .\end{aligned}$$

5° If $\phi = \mathbf{not}\,\psi$, then using the inductive hypothesis for ψ we obtain

$$\begin{aligned}(I, \mathcal{M}_1, \mathcal{N}) \models \phi &\iff \exists J \in \mathcal{N} : (J, \mathcal{M}_1, \mathcal{N}) \not\models \psi \\ &\iff \exists J \in \mathcal{N} : (J, \mathcal{M}_2, \mathcal{N}) \not\models \psi \iff (I, \mathcal{M}_2, \mathcal{N}) \models \phi \ .\end{aligned}$$

The second claim can be proved using the first one as follows:

$$(I, \mathcal{M}_1, \mathcal{N}) \models \mathcal{T} \iff \forall \phi \in \mathcal{T} : (I, \mathcal{M}_1, \mathcal{N}) \models \phi$$
$$\iff \forall \phi \in \mathcal{T} : (I, \mathcal{M}_2, \mathcal{N}) \models \phi \iff (I, \mathcal{M}_2, \mathcal{N}) \models \mathcal{T} \ . \qquad \square$$

Definition A.7 (Subjective Entailment)**.** Let ϕ be a subjective MKNF sentence and $\mathcal{T}$ a set of subjective MKNF sentences. For any $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ we write

$$(\mathcal{M}, \mathcal{N}) \models \phi \quad \text{if and only if} \quad \exists I \in \mathfrak{I} : (I, \mathcal{M}, \mathcal{N}) \models \phi \ ,$$
$$(\mathcal{M}, \mathcal{N}) \models \mathcal{T} \quad \text{if and only if} \quad \exists I \in \mathfrak{I} : (I, \mathcal{M}, \mathcal{N}) \models \mathcal{T} \ .$$

Proposition A.8 (Properties of Subjective Entailment)**.** *Let ϕ be a subjective MKNF sentence and $\mathcal{T}$ a set of subjective MKNF sentences. For all $I_1, I_2 \in \mathfrak{I}$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ it holds that*

(1) For all $I \in \mathfrak{I}$,

$$(\mathcal{M}, \mathcal{N}) \models \phi \iff (I, \mathcal{M}, \mathcal{N}) \models \phi \quad \textit{and} \quad (\mathcal{M}, \mathcal{N}) \models \mathcal{T} \iff (I, \mathcal{M}, \mathcal{N}) \models \mathcal{T} \ .$$

(2) If $\mathcal{M} \neq \emptyset$, then

$$(\mathcal{M}, \mathcal{M}) \models \phi \iff \mathcal{M} \models \phi \qquad \textit{and} \qquad (\mathcal{M}, \mathcal{M}) \models \mathcal{T} \iff \mathcal{M} \models \mathcal{T} \ .$$

(3) If $\mathcal{M} \neq \emptyset$ and both ϕ and $\mathcal{T}$ are **not***-free, then*

$$(\mathcal{M}, \mathcal{N}) \models \phi \iff \mathcal{M} \models \phi \qquad \textit{and} \qquad (\mathcal{M}, \mathcal{N}) \models \mathcal{T} \iff \mathcal{M} \models \mathcal{T} \ .$$

(4) If $\mathcal{N} \neq \emptyset$ and both ϕ and $\mathcal{T}$ are **K***-free, then*

$$(\mathcal{M}, \mathcal{N}) \models \phi \iff \mathcal{N} \models \phi \qquad \textit{and} \qquad (\mathcal{M}, \mathcal{N}) \models \mathcal{T} \iff \mathcal{N} \models \mathcal{T} \ .$$

(5) $(\mathcal{M}, \mathcal{N}) \models \mathcal{T}$ if and only if $(\mathcal{M}, \mathcal{N}) \models \psi$ for all $\psi \in \mathcal{T}$.

Proof. We consider each statement separately:

(1) This is a consequence of Lemma A.4.

(2) This is a consequence of claim (1).

(3) This is a consequence of claim (1) and Lemma A.5.

(4) This is a consequence of claim (1) and Lemma A.6.

(5) This is a consequence of claim (1). $\square$

Corollary A.9. *Let ψ, ψ_1, ψ_2 be subjective MKNF sentences and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. Then the following holds:*

$$(\mathcal{M}, \mathcal{N}) \models \neg\psi \quad \textit{if and only if} \quad (\mathcal{M}, \mathcal{N}) \not\models \psi$$
$$(\mathcal{M}, \mathcal{N}) \models \psi_1 \land \psi_2 \quad \textit{if and only if} \quad (\mathcal{M}, \mathcal{N}) \models \psi_1 \textit{ and } (\mathcal{M}, \mathcal{N}) \models \psi_2$$
$$(\mathcal{M}, \mathcal{N}) \models \psi_1 \supset \psi_2 \quad \textit{if and only if} \quad (\mathcal{M}, \mathcal{N}) \models \psi_1 \textit{ implies } (\mathcal{M}, \mathcal{N}) \models \psi_2 \ .$$

Proof. These are consequences of Proposition A.8(1). $\square$

Corollary A.10. *Let π be a ground MKNF rule and $\mathcal{M}$, $\mathcal{N}$ be MKNF interpretations. Then $\mathcal{M} \models \kappa(\pi)$ if and only if*

$$\mathcal{M} \models \bigwedge \kappa(B(\pi)) \qquad \textit{implies} \qquad \mathcal{M} \models \bigvee \kappa(H(\pi)) \ .$$

If $H(\pi)$ consists of a single generalised atom, then $(\mathcal{M}, \mathcal{N}) \models \kappa(\pi)$ if and only if

$$\mathcal{M} \models \kappa(B(\pi)^+) \text{ and } \mathcal{N} \models \kappa(\sim B(\pi)^-) \qquad \text{implies} \qquad \mathcal{M} \models \kappa(H(\pi)) \enspace .$$

If $H(\pi)$ consists of a single generalised default literal, then $(\mathcal{M}, \mathcal{N}) \models \kappa(\pi)$ if and only if

$$\mathcal{M} \models \kappa(B(\pi)^+) \text{ and } \mathcal{N} \models \kappa(\sim B(\pi)^-) \qquad \text{implies} \qquad \mathcal{N} \models \kappa(H(\pi)) \enspace .$$

Proof. The first claim follows from Proposition A.8(2) and Corollary A.9. The other two claims follow from Proposition A.8(2, 3, 4) and from Corollary A.9. □

Corollary A.11. *Let $\mathcal{K}$ be an MKNF knowledge base. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M} \models \kappa(\mathcal{K})$ and for all $\mathcal{M}' \supsetneq \mathcal{M}$, $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$.*

Proof. This follows from the fact that $\kappa(\mathcal{K})$ is a subjective theory and by applying Proposition A.8(1, 2). □

Proposition 2.21. *Let $\mathcal{K}$ be an MKNF knowledge base without default negation in heads of rules. If $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, then $\mathcal{M}$ is a subset-maximal S5 model of $\mathcal{K}$.*

Proof. Suppose that $\mathcal{M}$ and $\mathcal{M}'$ are MKNF models of an MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ and $\mathcal{M} \subseteq \mathcal{M}'$. We will show that $(\mathcal{M}', \mathcal{M}) \models \kappa(\mathcal{K})$, which, together with Corollary A.11, implies that $\mathcal{M}' = \mathcal{M}$. Take some formula $\phi \in \kappa(\mathcal{K})$. Since $\mathcal{M}'$ is an MKNF model of $\mathcal{K}$, it must be the case that

$$(\mathcal{M}', \mathcal{M}') \models \phi \enspace . \tag{A.2}$$

If $\phi = \mathbf{K}\, \kappa(\psi)$ for some $\psi \in \mathcal{O}$, then ϕ is **not**-free and it follows from (A.2) by Proposition A.8(3) that $(\mathcal{M}', \mathcal{M}) \models \phi$.

The other case occurs when $\phi = \bigwedge \kappa(B(\pi)) \supset \bigvee \kappa(H(\pi))$ for some $\pi \in \mathcal{P}$. By Corollary A.10, assuming that $(\mathcal{M}', \mathcal{M}) \models \bigwedge \kappa(B(\pi))$, we need to prove that $(\mathcal{M}', \mathcal{M}) \models \bigvee \kappa(H(\pi))$. First we are going to show that

$$(\mathcal{M}', \mathcal{M}') \models \bigwedge \kappa(B(\pi)) \enspace . \tag{A.3}$$

Take some $L \in B(\pi)$. If L is a generalised atom ξ, then it follows that $\kappa(\xi) = \mathbf{K}\, \xi$ is **not**-free and it follows that $(\mathcal{M}', \mathcal{M}') \models \mathbf{K}\, \xi$. On the other hand, if L is a generalised literal $\sim\xi$, then $\kappa(L) = \mathbf{not}\, \xi$ and we obtain that $\mathcal{M} \not\models \xi$. But since $\mathcal{M} \subseteq \mathcal{M}'$, we immediately obtain that $\mathcal{M}' \not\models \xi$ and it follows that $(\mathcal{M}', \mathcal{M}') \models \mathbf{not}\, \xi$. This establishes (A.3).

From (A.3) and (A.2) we can infer that

$$(\mathcal{M}', \mathcal{M}') \models \bigvee \kappa(H(\pi)) \enspace .$$

By assumption, $H(\pi)$ contains no default generalised literals, so $(\mathcal{M}', \mathcal{M}') \models \mathbf{K}\, \xi$ for some generalised atom $\xi \in H(\pi)$. And since $\mathbf{K}\, \xi$ is **not**-free, we conclude that $(\mathcal{M}', \mathcal{M}) \models \bigvee \kappa(H(\pi))$, as we wanted to show. □

Proposition 2.22. *Let $\mathcal{K}$ be an MKNF knowledge base and $\mathcal{K}'$ the MKNF knowledge base obtained from $\mathcal{K}$ by replacing every non-disjunctive rule with default negation in the head*

$$\sim\xi \leftarrow B^+, \sim B^-. \qquad \text{with the rule} \qquad \leftarrow \xi, B^+, \sim B^-.$$

Then the MKNF models of $\mathcal{K}$ coincide with the MKNF models of $\mathcal{K}'$.

Proof. Take an MKNF model $\mathcal{M}$ of $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. We will show that $\mathcal{M}$ is an MKNF model of $\mathcal{K}'$. We first need to show that $\mathcal{M}$ is an S5 model of $\mathcal{K}' = (\mathcal{O}, \mathcal{P}')$. Take some $\phi \in \kappa(\mathcal{K}')$. If ϕ belongs to $\kappa(\mathcal{K})$, then certainly $\mathcal{M} \models \phi$. So suppose that $\phi = \kappa(\pi')$ for some replacement $\pi' \in \mathcal{P}'$ of a rule $\pi \in \mathcal{P}$. Then either $\mathcal{M} \not\models \bigwedge \kappa(B(\pi))$ or $\mathcal{M} \models \mathbf{not}\, \xi$. In either case, $\mathcal{M} \not\models \bigwedge \kappa(B(\pi'))$, so $\mathcal{M} \models \kappa(\pi')$ as desired.

Now suppose that $\mathcal{M}' \supsetneq \mathcal{M}$. Since $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, there is a formula $\phi \in \kappa(\mathcal{K})$ such that,

$$(\mathcal{M}', \mathcal{M}) \not\models \phi \tag{A.4}$$

We will prove by contradiction that ϕ also belongs to $\kappa(\mathcal{K}')$ which establishes that $\mathcal{M}$ is an MKNF model of $\mathcal{K}'$. So suppose that ϕ does not belong to $\kappa(\mathcal{K}')$. Then,

$$\phi = \left(\bigwedge \kappa(B(\pi)) \supset \mathbf{not}\, \xi\right)$$

and it follows from (A.4) that $(\mathcal{M}', \mathcal{M}) \models \bigwedge \kappa(B(\pi))$ and $(\mathcal{M}', \mathcal{M}) \not\models \mathbf{not}\, \xi$. The former implies that $(\mathcal{M}, \mathcal{M}) \models \bigwedge \kappa(B(\pi))$ and since $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, we conclude that $(\mathcal{M}, \mathcal{M}) \models \mathbf{not}\, \xi$. But then it follows that $(\mathcal{M}', \mathcal{M}) \models \mathbf{not}\, \xi$, contrary to our previous conclusion.

For the other direction, suppose that $\mathcal{M}$ is an MKNF model of $\mathcal{K}'$. To verify that $\mathcal{M}$ is an S5 model of $\mathcal{K}$, take some $\phi \in \kappa(\mathcal{K})$. If ϕ belongs to $\kappa(\mathcal{K}')$, then certainly $\mathcal{M} \models \phi$. So suppose that $\phi = \kappa(\pi)$ for some rule π replaced in $\mathcal{K}'$ by π'. It follows that $\mathcal{M} \not\models \bigwedge \kappa(B(\pi'))$, which implies that either $\mathcal{M} \not\models \bigwedge \kappa(B(\pi))$ or $\mathcal{M} \not\models \xi$. In either case we can conclude that $\mathcal{M} \models \kappa(\pi)$ as desired.

Finally, assume that $\mathcal{M}' \supsetneq \mathcal{M}$. Since $\mathcal{M}$ is an MKNF model of $\mathcal{K}'$, there is a formula $\phi' \in \kappa(\mathcal{K}')$ such that

$$(\mathcal{M}', \mathcal{M}) \not\models \phi' \tag{A.5}$$

If ϕ' belongs to $\kappa(\mathcal{K})$, then it follows that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ and the proof is finished. Otherwise we obtain

$$\phi' = \left(\bigwedge \kappa(B(\pi) \cup \{\,\xi\,\}) \supset \bot\right) \tag{A.6}$$

and $\kappa(\mathcal{K})$ contains the formula $\phi = (\bigwedge \kappa(B(\pi)) \supset \mathbf{not}\, \xi)$. It is not difficult to verify that (A.5) and (A.6) imply that $(\mathcal{M}', \mathcal{M}) \models \bigwedge \kappa(B(\pi))$ and $\mathcal{M}' \models \xi$, so it follows from $\mathcal{M}' \supseteq \mathcal{M}$ that $\mathcal{M} \models \xi$, implying that $(\mathcal{M}', \mathcal{M}) \not\models \mathbf{not}\, \xi$ and thus $(\mathcal{M}', \mathcal{M}) \not\models \phi$, which proves that $\mathcal{M}$ is indeed an MKNF model of $\mathcal{K}$. □

A.2.3 Restricted MKNF Interpretations

This section contains proofs of the essential properties of *interpretation restrictions*. This concept is formally defined in Definition 2.39 on page 38.

Definition A.12 (Interpretation Coincidence)**.** Let A be a set of predicate symbols. For any $I, J \in \mathfrak{I}$ we say that I *coincides with* J *on* A if $I^{[A]} = J^{[A]}$.

Similarly, for any $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$, $\mathcal{M}$ *coincides with* $\mathcal{N}$ *on* A if $\mathcal{M}^{[A]} = \mathcal{N}^{[A]}$.

Proposition A.13. *Let ϕ be an MKNF sentence, A a set of predicate symbols such that $A \supseteq \mathsf{pr}(\phi)$ and $(I, \mathcal{M}, \mathcal{N})$ an MKNF structure. Then:*

$$(I, \mathcal{M}, \mathcal{N}) \models \phi \iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ .$$

Proof. We prove by structural induction on ϕ:

1° If ϕ is a ground atom p, then the following chain of equivalences proves the claim:

$$(I, \mathcal{M}, \mathcal{N}) \models \phi \iff I \models p \iff I^{[A]} \models p \iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ ;$$

2° If ϕ is of the form $\neg\psi$, then $\mathsf{pr}(\phi) = \mathsf{pr}(\psi)$, so $A \supseteq \mathsf{pr}(\psi)$. Hence, we can use the inductive hypothesis for ψ as follows:

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}) \models \phi \iff (I, \mathcal{M}, \mathcal{N}) \not\models \psi &\iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \not\models \psi \\ &\iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ ;\end{aligned}$$

3° If ϕ is of the form $\phi_1 \wedge \phi_2$, then $\mathsf{pr}(\phi) = \mathsf{pr}(\phi_1) \cup \mathsf{pr}(\phi_2)$, so we obtain both $A \supseteq \mathsf{pr}(\phi_1)$ and $A \supseteq \mathsf{pr}(\phi_2)$. Applying the inductive hypothesis to ϕ_1 and ϕ_2 now yields the claim:

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}) \models \phi &\iff (I, \mathcal{M}, \mathcal{N}) \models \phi_1 \wedge (I, \mathcal{M}, \mathcal{N}) \models \phi_2 \\ &\iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi_1 \wedge \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi_2 \\ &\iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ ;\end{aligned}$$

4° If ϕ is of the form $\exists \mathbf{x} : \psi$, then for all $a \in \mathcal{C}$, $\mathsf{pr}(\phi) = \mathsf{pr}(\psi) = \mathsf{pr}(\psi[a/\mathbf{x}])$, so $A \supseteq \mathsf{pr}(\psi[a/\mathbf{x}])$. Consequently, we can use the inductive hypothesis for the formulae $\psi[a/\mathbf{x}]$ as follows:

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}) \models \phi &\iff \exists a \in \mathcal{C} : (I, \mathcal{M}, \mathcal{N}) \models \psi[a/\mathbf{x}] \\ &\iff \exists a \in \mathcal{C} : \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \psi[a/\mathbf{x}] \\ &\iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ ;\end{aligned}$$

5° If ϕ is of the form $\mathbf{K}\,\psi$, then $\mathsf{pr}(\phi) = \mathsf{pr}(\psi)$, so $A \supseteq \mathsf{pr}(\psi)$. The claim now follows from the inductive hypothesis for ψ:

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}) \models \phi &\iff \forall J \in \mathcal{M} : (J, \mathcal{M}, \mathcal{N}) \models \psi \\ &\iff \forall J \in \mathcal{M} : \left(J^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \psi \\ &\iff \forall J \in \mathcal{M}^{[A]} : (J, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}) \models \psi \\ &\iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ ;\end{aligned}$$

6° If ϕ is of the form $\mathbf{not}\,\psi$, then $\mathsf{pr}(\phi) = \mathsf{pr}(\psi)$, so $A \supseteq \mathsf{pr}(\psi)$. The claim follows similarly as in the previous case:

$$\begin{aligned}(I, \mathcal{M}, \mathcal{N}) \models \phi &\iff \exists J \in \mathcal{N} : (J, \mathcal{M}, \mathcal{N}) \not\models \psi \\ &\iff \exists J \in \mathcal{N} : \left(J^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \not\models \psi \\ &\iff \exists J \in \mathcal{N}^{[A]} : (J, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}) \not\models \psi \\ &\iff \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ .\end{aligned}$$

□

Corollary A.14. *Let $\mathcal{T}$ be an MKNF theory, A a set of predicate symbols such that $A \supseteq \mathsf{pr}(\mathcal{T})$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ be such that they coincide on A. Then,*

$$\mathcal{M} \models \mathcal{T} \qquad \textit{if and only if} \qquad \mathcal{N} \models \mathcal{T} \ .$$

Proof. We prove only the direct implication, the proof of the converse direction follows from the symmetry of the claim. Suppose that $\mathcal{M} \models \mathcal{T}$. Then for every $\phi \in \mathcal{T}$ and all $I \in \mathcal{M}$ we have $(I, \mathcal{M}, \mathcal{M}) \models \phi$. We want to show that $\mathcal{N} \models \mathcal{T}$. Pick some $\psi \in \mathcal{T}$ and some $J \in \mathcal{N}$. Since $\mathcal{M}^{[A]} = \mathcal{N}^{[A]}$, there must be some $I \in \mathcal{M}$ such that $I^{[A]} = J^{[A]}$. By assumption, $A \supseteq \mathsf{pr}(\psi)$ and $(I, \mathcal{M}, \mathcal{M}) \models \psi$, so Proposition A.13 yields $\left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{M}^{[A]}\right) \models \psi$. From $I^{[A]} = J^{[A]}$ and $\mathcal{M}^{[A]} = \mathcal{N}^{[A]}$ it thus follows that $\left(J^{[A]}, \mathcal{N}^{[A]}, \mathcal{N}^{[A]}\right) \models \psi$. By applying Proposition A.13 once again we obtain $(J, \mathcal{N}, \mathcal{N}) \models \psi$ and, thus, $\mathcal{N} \models \mathcal{T}$. □

Proposition A.15. *Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. Then the following holds:*

(1) If $\mathcal{M} \subseteq \mathcal{N}$, then $\mathcal{M}^{[A]} \subseteq \mathcal{N}^{[A]}$.

(2) If $\mathcal{M} = \mathcal{N}$, then $\mathcal{M}^{[A]} = \mathcal{N}^{[A]}$.

(3) If $\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]}$, then $\mathcal{M} \subsetneq \mathcal{N}$.

Proof.

(1) Follows by definition of $\mathcal{M}^{[A]}$ and $\mathcal{N}^{[A]}$.

(2) This is a direct consequence of (1).

(3) Suppose that $\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]}$. Then there is some $J \in \mathcal{N}$ such that $J^{[A]} \notin \mathcal{M}^{[A]}$. Consequently, $J \notin \mathcal{M}$ and it follows that $\mathcal{M}$ is a proper subset of $\mathcal{N}$. □

Proposition A.16. *Let ϕ be a subjective MKNF sentence, A a set of predicate symbols such that $A \supseteq \mathsf{pr}(\phi)$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. Then,*

$$(\mathcal{M}, \mathcal{N}) \models \phi \qquad \textit{if and only if} \qquad \left(\mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ .$$

Proof. By Definition A.7 we have

$$(\mathcal{M}, \mathcal{N}) \models \phi \qquad \text{if and only if} \qquad \exists I \in \mathfrak{I} : (I, \mathcal{M}, \mathcal{N}) \models \phi \ .$$

By Proposition A.13 we can equivalently rewrite the right hand side into

$$\exists I \in \mathfrak{I} : \left(I^{[A]}, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ .$$

Furthermore, since ϕ is subjective, we can use Lemma A.4 to further rewrite the previous statement into

$$\exists I \in \mathfrak{I} : \left(I, \mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ ,$$

which is by Definition A.7 equivalent to

$$\left(\mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \ .$$ □

Corollary A.17. *Let ϕ be a subjective MKNF sentence, A a set of predicate symbols such that $A \supseteq \mathsf{pr}(\phi)$ and $\mathcal{M}, \mathcal{M}', \mathcal{N}, \mathcal{N}' \in \mathfrak{M}$ be such that $\mathcal{M}$ coincides with $\mathcal{M}'$ on A and $\mathcal{N}$ coincides* *with $\mathcal{N}'$ on A. Then,*

$$(\mathcal{M}, \mathcal{N}) \models \phi \iff (\mathcal{M}', \mathcal{N}') \models \phi \ .$$

Proof. By assumptions we know that $\mathcal{M}^{[A]} = \mathcal{M}'^{[A]}$ and $\mathcal{N}^{[A]} = \mathcal{N}'^{[A]}$. Proposition A.16 now yields:

$$(\mathcal{M}, \mathcal{N}) \models \phi \iff \left(\mathcal{M}^{[A]}, \mathcal{N}^{[A]}\right) \models \phi \iff \left(\mathcal{M}'^{[A]}, \mathcal{N}'^{[A]}\right) \models \phi \iff (\mathcal{M}', \mathcal{N}') \models \phi \ . \quad \square$$

A.2.4 Saturated MKNF Interpretations

The statements and proofs in the present as well as the next two sections assume that the use of the equality predicate $\approx$ is not allowed.

For a definition of a *saturated MKNF interpretation*, please refer to Definition 2.40 on page 38.

We start by showing that every MKNF model of an MKNF theory $\mathcal{T}$ is saturated relative to the set of predicate symbols relevant to $\mathcal{T}$:

Proposition A.18. *Let A be a set of predicate symbols and $\mathcal{T}$ an MKNF theory such that $A \supseteq \mathsf{pr}(\mathcal{T})$. If $\mathcal{M}$ is an MKNF model of $\mathcal{T}$, then $\mathcal{M}$ is saturated relative to A.*

Proof. Suppose that $\mathcal{M}$ is not saturated relative to A. Then there is some $I \in \mathfrak{I}$ such that $I^{[A]} \in \mathcal{M}^{[A]}$ and $I \notin \mathcal{M}$. Let $\mathcal{M}' = \mathcal{M} \cup \{ I \}$. Since $\mathcal{M}$ is an MKNF model of $\mathcal{T}$, it follows that $(I', \mathcal{M}', \mathcal{M}) \not\models \mathcal{T}$ for some $I' \in \mathcal{M}'$. But $I'^{[A]} \in \mathcal{M}'^{[A]} = \mathcal{M}^{[A]}$, so there must be some $I'' \in \mathcal{M}$ such that $I''^{[A]} = I'^{[A]}$. By two applications of Proposition A.13 we obtain

$$(I', \mathcal{M}', \mathcal{M}) \not\models \mathcal{T} \implies (I'^{[A]}, \mathcal{M}'^{[A]}, \mathcal{M}^{[A]}) \not\models \mathcal{T} \implies (I'', \mathcal{M}, \mathcal{M}) \not\models \mathcal{T} \ .$$

This is in conflict with the assumption that $\mathcal{M}$ is an MKNF model of $\mathcal{T}$. $\square$

Similarly, the set of models of a first-order theory $\mathcal{T}$ is saturated relative to the set of predicate symbols relevant to $\mathcal{T}$.

Proposition 2.41. *Let A be a set of predicate symbols and $\mathcal{T}$ a first-order theory such that $\mathsf{pr}(\mathcal{T}) \subseteq A$. Then $[\![\mathcal{T}]\!]$ is saturated relative to A.*

Proof. If $[\![\mathcal{T}]\!] = \emptyset$, then this follows from the fact that $\emptyset$ is trivially saturated relative to any set of predicate symbols. Otherwise, $[\![\mathcal{T}]\!]$ is the MKNF model of $\mathcal{T}$ and it suffices to use Proposition A.18. $\square$

Two strengthened versions of Proposition A.15 can be shown for saturated MKNF interpretations, as introduced in Definition 2.40, with implications replaced by equivalences.

Proposition A.19. *Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ such that $\mathcal{M}$ is saturated relative to A. Then the following holds:*

(1) $\mathcal{M} = \mathcal{N}$ if and only if $\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{M}^{[A]} \supseteq \mathcal{N}^{[A]}$.

(2) $\mathcal{M} \subsetneq \mathcal{N}$ if and only if $\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]}$.

Proof.

(1) The direct implication follows from Proposition A.15. To prove the converse implication, suppose that $\mathcal{M}^{[A]} \supseteq \mathcal{N}^{[A]}$ and $I \in \mathcal{N}$. We immediately obtain $I^{[A]} \in \mathcal{M}^{[A]}$ and since $\mathcal{M}$ is saturated relative to A, we can conclude that $I \in \mathcal{M}$.

(2) For the direct implication suppose that $\mathcal{M} \subsetneq \mathcal{N}$. Then there is some $I \in \mathcal{N}$ such that $I \notin \mathcal{M}$. Since $\mathcal{M}$ is saturated relative to A, we obtain that $I^{[A]} \notin \mathcal{M}^{[A]}$. Consequently, $\mathcal{M}^{[A]}$ is a proper subset of $\mathcal{N}^{[A]}$. The converse implication is a consequence of Proposition A.15(3). □

Proposition A.20. *Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ such that $\mathcal{N}$ is saturated relative to A. Then the following holds:*

(1) $\mathcal{M} \subseteq \mathcal{N}$ if and only if $\mathcal{M}^{[A]} \subseteq \mathcal{N}^{[A]}$.

(2) $\mathcal{M} = \mathcal{N}$ if and only if $\mathcal{M}^{[A]} = \mathcal{N}^{[A]}$.

(3) $\mathcal{M} \subsetneq \mathcal{N}$ if and only if $\mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]}$.

Proof.

(1) The direct implication follows from Proposition A.15(1). To prove the converse implication, suppose that $\mathcal{M}^{[A]} \subseteq \mathcal{N}^{[A]}$ and $I \in \mathcal{M}$. We immediately obtain $I^{[A]} \in \mathcal{M}^{[A]}$, hence also $I^{[A]} \in \mathcal{N}^{[A]}$. Since $\mathcal{N}$ is saturated relative to A, we can conclude that $I \in \mathcal{N}$. Consequently, $\mathcal{M} \subseteq \mathcal{N}$.

(2) This is a consequence of (1).

(3) This is a consequence of (1) and (2). □

The following concept and its properties, strongly related to saturated interpretations, will be essential for proving all our results in Chapter 4.

Definition A.21. Let A be a set of predicate symbols and $\mathcal{M} \in \mathfrak{M}$. We introduce the following notation:

$$\sigma(\mathcal{M}, A) = \left\{ I \in \mathfrak{I} \;\middle|\; I^{[A]} \in \mathcal{M}^{[A]} \right\}$$

Proposition A.22. *Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. Then the following conditions are equivalent:*

1. $\mathcal{N} = \sigma(\mathcal{M}, A)$;

2. $\mathcal{N}$ coincides with $\mathcal{M}$ on A and is saturated relative to A.

3. $\mathcal{N}$ is the greatest among all $\mathcal{N}' \in \mathfrak{M}$ coinciding with $\mathcal{M}$ on A;

Furthermore, if $\mathcal{N}$ satisfies one of the conditions above, then $\mathcal{M} \subseteq \mathcal{N}$.

Proof. We will prove that 1. implies 2., 2. implies 3. and finally that 3. implies 1.

Suppose that $\mathcal{N} = \sigma(\mathcal{M}, A)$. Then,

$$\mathcal{N}^{[A]} = \{ I^{[A]} \mid I \in \mathfrak{I} \wedge I^{[A]} \in \mathcal{M}^{[A]} \} = \mathcal{M}^{[A]} \ ,$$

so $\mathcal{N}$ coincides with $\mathcal{M}$ on A. Furthermore, any $I \in \mathfrak{I}$ with $I^{[A]} \in \mathcal{N}^{[A]}$ must also satisfy $I^{[A]} \in \mathcal{M}^{[A]}$ and by the definition of $\sigma(\mathcal{M}, A)$ we obtain $I \in \mathcal{N}$, so $\mathcal{N}$ is saturated relative to A. This shows that 1. implies 2.

To prove that 2. implies 3., suppose that $\mathcal{N}$ coincides with $\mathcal{M}$ on A and is saturated relative to A. Take some $\mathcal{N}' \in \mathfrak{M}$ that coincides with $\mathcal{M}$ on A and $I \in \mathcal{N}'$. Then,

$$I^{[A]} \in \mathcal{N}'^{[A]} = \mathcal{M}^{[A]} = \mathcal{N}^{[A]} \ .$$

Since $\mathcal{N}$ is saturated relative to A, we can conclude that I belongs to $\mathcal{N}$. Consequently, $\mathcal{N}'$ is included in $\mathcal{N}$, so $\mathcal{N}$ is the greatest among all $\mathcal{N}' \in \mathfrak{M}$ coinciding with $\mathcal{M}$ on A.

Finally, suppose that $\mathcal{N}$ is the greatest among all $\mathcal{N}' \in \mathfrak{M}$ coinciding with $\mathcal{M}$ on A. It follows that since $\sigma(\mathcal{M}, A)$ coincides with $\mathcal{M}$ on A, it must be a subset of $\mathcal{N}$. It remains to show that $\mathcal{N}$ is a subset of $\sigma(\mathcal{M}, A)$. But that is a consequence of the fact that for any $I \in \mathcal{N}$, $I^{[A]}$ belongs to $\mathcal{M}^{[A]}$ and, thus, I belongs to $\sigma(\mathcal{M}, A)$.

It remains to show that $\mathcal{M}$ is a subset of $\mathcal{N}$ if $\mathcal{N}$ satisfies one of the above conditions. We already know that the conditions are equivalent, so we only need to consider one of them. So suppose that $\mathcal{N} = \sigma(\mathcal{M}, A)$. It follows from the definition of $\sigma(\mathcal{M}, A)$ that every $I \in \mathcal{M}$ belongs to $\mathcal{N}$ as well. Hence, $\mathcal{M}$ is a subset of $\mathcal{N}$. □

Proposition A.23. *Let A_1, A_2 be sets of predicate symbols and $\mathcal{M} \in \mathfrak{M}$. Then,*

$$\sigma(\sigma(\mathcal{M}, A_1), A_2) = \sigma(\mathcal{M}, A_1 \cap A_2) \ .$$

Proof. Consider the following sequence of equivalences:

$$\begin{aligned} I \in \sigma(\sigma(\mathcal{M}, A_1), A_2) &\iff I^{[A_2]} \in \sigma(\mathcal{M}, A_1)^{[A_2]} \\ &\iff \exists J \in \sigma(\mathcal{M}, A_1) : J^{[A_2]} = I^{[A_2]} \\ &\iff \exists J \in \mathfrak{I}\ \exists K \in \mathcal{M} : K^{[A_1]} = J^{[A_1]} \wedge J^{[A_2]} = I^{[A_2]} \\ &\iff \exists K \in \mathcal{M}\ \exists J \in \mathfrak{I} : J^{[A_1]} = K^{[A_1]} \wedge J^{[A_2]} = I^{[A_2]} \ . \end{aligned}$$

Moreover, we also obtain the following:

$$I \in \sigma(\mathcal{M}, A_1 \cap A_2) \iff \exists K \in \mathcal{M} : K^{[A_1 \cap A_2]} = I^{[A_1 \cap A_2]} \ .$$

So it remains to show that

$$\exists J \in \mathfrak{I} : J^{[A_1]} = K^{[A_1]} \wedge J^{[A_2]} = I^{[A_2]} \qquad \text{if and only if} \qquad K^{[A_1 \cap A_2]} = I^{[A_1 \cap A_2]} \ .$$

Indeed, if such a J exists, then for every ground atom p the following holds:

$$\begin{aligned} K^{[A_1 \cap A_2]} \models p &\iff K \models p \wedge \mathsf{pr}(p) \subseteq A_1 \cap A_2 \\ &\iff (K \models p \wedge \mathsf{pr}(p) \subseteq A_1) \wedge \mathsf{pr}(p) \subseteq A_2 \\ &\iff K^{[A_1]} \models p \wedge \mathsf{pr}(p) \subseteq A_2 \\ &\iff J^{[A_1]} \models p \wedge \mathsf{pr}(p) \subseteq A_2 \\ &\iff (J \models p \wedge \mathsf{pr}(p) \subseteq A_2) \wedge \mathsf{pr}(p) \subseteq A_1 \\ &\iff J^{[A_2]} \models p \wedge \mathsf{pr}(p) \subseteq A_1 \\ &\iff I^{[A_2]} \models p \wedge \mathsf{pr}(p) \subseteq A_1 \\ &\iff I \models p \wedge \mathsf{pr}(p) \subseteq A_1 \cap A_2 \\ &\iff I^{[A_1 \cap A_2]} \models p \end{aligned}$$

On the other hand, if the other condition holds, then let J be an interpretation such that for every ground atom p,

$$J \models p \qquad \text{if and only if} \qquad (K \models p \wedge \mathsf{pr}(p) \subseteq A_1) \vee (I \models p \wedge \mathsf{pr}(p) \subseteq A_2)$$

For any ground atom p we obtain:

$$\begin{aligned} J^{[A_1]} \models p &\iff J \models p \wedge \mathsf{pr}(p) \subseteq A_1 \\ &\iff (K \models p \wedge \mathsf{pr}(p) \subseteq A_1) \vee (I \models p \wedge \mathsf{pr}(p) \subseteq A_1 \cap A_2) \\ &\iff (K \models p \wedge \mathsf{pr}(p) \subseteq A_1) \vee I^{[A_1 \cap A_2]} \models p \end{aligned}$$

$$
\begin{aligned}
&\iff (K \models p \wedge \mathsf{pr}(p) \subseteq A_1) \vee K^{[A_1 \cap A_2]} \models p \\
&\iff (K \models p \wedge \mathsf{pr}(p) \subseteq A_1) \vee (K \models p \wedge \mathsf{pr}(p) \subseteq A_1 \cap A_2) \\
&\iff K \models p \wedge \mathsf{pr}(p) \subseteq A_1 \\
&\iff K^{[A_1]} \models p
\end{aligned}
$$

and also

$$
\begin{aligned}
J^{[A_2]} \models p &\iff J \models p \wedge \mathsf{pr}(p) \subseteq A_2 \\
&\iff (K \models p \wedge \mathsf{pr}(p) \subseteq A_1 \cap A_2) \vee (I \models p \wedge \mathsf{pr}(p) \subseteq A_2) \\
&\iff K^{[A_1 \cap A_2]} \models p \vee (I \models p \wedge \mathsf{pr}(p) \subseteq A_2) \\
&\iff I^{[A_1 \cap A_2]} \models p \vee (I \models p \wedge \mathsf{pr}(p) \subseteq A_2) \\
&\iff (I \models p \wedge \mathsf{pr}(p) \subseteq A_1 \cap A_2) \vee (I \models p \wedge \mathsf{pr}(p) \subseteq A_2) \\
&\iff I \models p \wedge \mathsf{pr}(p) \subseteq A_2 \\
&\iff I^{[A_2]} \models p \ .
\end{aligned}
$$

□

Proposition A.24. *Let A_1, A_2 be sets of predicate symbols such that $A_1 \subseteq A_2$ and $\mathcal{M} \in \mathfrak{M}$. Then,*

$$\sigma(\mathcal{M}, A_2)^{[A_1]} = \mathcal{M}^{[A_1]} \ .$$

Proof. First suppose that $I \in \sigma(\mathcal{M}, A_2)^{[A_1]}$. Then for some $J \in \sigma(\mathcal{M}, A_2)$ we have $I = J^{[A_1]}$, so for every ground atom p,

$$I \models p \qquad \text{if and only if} \qquad J \models p \wedge \mathsf{pr}(p) \subseteq A_1 \ . \tag{A.7}$$

Also, since J belongs to $\sigma(\mathcal{M}, A_2)$, there must be some $K \in \mathcal{M}$ such that $K^{[A_2]} = J^{[A_2]}$, which means that for any ground atom p with $\mathsf{pr}(p) \subseteq A_2$ we have $J \models p \iff K \models p$. This, together with (A.7) and the assumption that A_1 is a subset of A_2, implies that for every ground atom p,

$$I \models p \qquad \text{if and only if} \qquad K \models p \wedge \mathsf{pr}(p) \subseteq A_1 \ .$$

Thus, $I \in \mathcal{M}^{[A_1]}$.

The converse inclusion follows from the fact that $\mathcal{M}$ is a subset of $\sigma(\mathcal{M}, A_2)$ (see Proposition A.22). □

Lemma A.25. *Let A_1, A_2 be sets of predicate symbols such that $A_1 \subseteq A_2$ and $\mathcal{M} \in \mathfrak{M}$. If $\mathcal{M}$ is saturated relative to A_1, then it is also saturated relative to A_2.*

Proof. Suppose that $\mathcal{M}$ is saturated relative to A_1 and $I \in \mathfrak{I}$ is such that $I^{[A_2]} \in \mathcal{M}^{[A_2]}$. We need to prove that $I \in \mathcal{M}$. We know that for some $J \in \mathcal{M}$ it holds that $I^{[A_2]} = J^{[A_2]}$. In other words, for every ground atom p with $\mathsf{pr}(p) \subseteq A_2$ it holds that:

$$I \models p \qquad \text{if and only if} \qquad J \models p \ .$$

Since A_1 is a subset of A_2, every ground atom p with $\mathsf{pr}(p) \subseteq A_1$ also satisfies the above equivalence. Thus, $I^{[A_1]} = J^{[A_1]}$ and we conclude that $I^{[A_1]} \in \mathcal{M}^{[A_1]}$. Since $\mathcal{M}$ is saturated relative to A_1, it follows that $I \in \mathcal{M}$. □

Lemma A.26. *Let A, B be disjoint sets of predicate symbols and $\mathcal{M}$ an MKNF interpretation. Then,*

$$\sigma(\mathcal{M}, A)^{[B]} = \mathcal{I}^{[B]} \quad .$$

Proof. Since $\sigma(\mathcal{M}, A)$ is a subset of $\mathcal{I}$, the left to right inclusion holds trivially. Suppose that $I \in \mathcal{I}^{[B]}$, i.e. $I \models p$ only if $\mathsf{pr}(p) \subseteq B$. Furthermore, take some $I' \in \mathcal{M}$ and let I'' be an interpretation such that $I'' \models p$ if and only if $I'^{[A]} \models p$ or $I \models p$. Since A is disjoint from B, it follows that $I''^{[A]} = I'^{[A]}$ and $I''^{[B]} = I$. This implies that I'' belongs to $\sigma(\mathcal{M}, A)$, so I belongs to $\sigma(\mathcal{M}, A)^{[B]}$. □

A.2.5 Semi-saturated MKNF Interpretations

There is also another class of MKNF interpretations for which a slightly modified version of Proposition A.15 holds. We introduce it here and look at some of its properties. These are useful in the proofs of the splitting set theorems in Appendix C.

Definition A.27 (Semi-saturated MKNF Interpretation)**.** Let A be a set of predicate symbols and $\mathcal{M} \in \mathfrak{M}$. We say that $\mathcal{M}$ is *semi-saturated relative to A* if for every interpretation $I \in \mathcal{I}$,

$$I^{[A]} \in \mathcal{M}^{[A]} \land I^{[\mathcal{P} \setminus A]} \in \mathcal{M}^{[\mathcal{P} \setminus A]} \qquad \text{implies} \qquad I \in \mathcal{M} \quad .$$

Proposition A.28. *Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N}$ be MKNF interpretations such that $\mathcal{M}$ is saturated relative to A and $\mathcal{N}$ is saturated relative to $\mathcal{P} \setminus A$. Then $\mathcal{M} \cap \mathcal{N}$ is an MKNF interpretation that is semi-saturated relative to A and coincides with $\mathcal{M}$ on A and with $\mathcal{N}$ on $\mathcal{P} \setminus A$.*

Proof. We first prove the following claim: For every $I \in \mathcal{M}$ and every $J \in \mathcal{N}$ there exists some $K \in \mathcal{M} \cap \mathcal{N}$ such that $K^{[A]} = I^{[A]}$ and $K^{[\mathcal{P} \setminus A]} = J^{[\mathcal{P} \setminus A]}$. The reason this holds is that the sets A and $(\mathcal{P} \setminus A)$ are disjoint. Let's take some $I \in \mathcal{M}$ and some $J \in \mathcal{N}$ and let K be an interpretation such that for every ground atom p,

$$K \models p \qquad \text{if and only if} \qquad I^{[A]} \models p \lor J^{[\mathcal{P} \setminus A]} \models p \quad .$$

The following can now be derived:

$$\begin{aligned}
K^{[A]} \models p &\iff K \models p \land \mathsf{pr}(p) \subseteq A \\
&\iff I \models p \land \mathsf{pr}(p) \subseteq A \lor J \models p \land \mathsf{pr}(p) \subseteq A \cap (\mathcal{P} \setminus A) \\
&\iff I \models p \land \mathsf{pr}(p) \subseteq A \\
&\iff I^{[A]} \models p \enspace , \\
K^{[\mathcal{P} \setminus A]} \models p &\iff K \models p \land \mathsf{pr}(p) \subseteq \mathcal{P} \setminus A \\
&\iff I \models p \land \mathsf{pr}(p) \subseteq A \cap (\mathcal{P} \setminus A) \lor J \models p \land \mathsf{pr}(p) \subseteq \mathcal{P} \setminus A \\
&\iff J \models p \land \mathsf{pr}(p) \subseteq \mathcal{P} \setminus A \\
&\iff J^{[\mathcal{P} \setminus A]} \models p \enspace .
\end{aligned}$$

Consequently, K belongs to $\mathcal{M}$ and also to $\mathcal{N}$ (because they are saturated relative to A and $\mathcal{P} \setminus A$, respectively), so K belongs to $\mathcal{M} \cap \mathcal{N}$ as well.

It follows from the above that $\mathcal{M} \cap \mathcal{N}$ is non-empty and that $(\mathcal{M} \cap \mathcal{N})^{[A]} = \mathcal{M}^{[A]}$ as well as $(\mathcal{M} \cap \mathcal{N})^{[\mathcal{P} \setminus A]} = \mathcal{N}^{[\mathcal{P} \setminus A]}$.

It remains to show that $\mathcal{M} \cap \mathcal{N}$ is semi-saturated relative to A. Let $I \in \mathcal{I}$ be such that $I^{[A]} \in (\mathcal{M} \cap \mathcal{N})^{[A]}$ and $I^{[\mathcal{P} \setminus A]} \in (\mathcal{M} \cap \mathcal{N})^{[\mathcal{P} \setminus A]}$. We need to prove that $I \in \mathcal{M} \cap \mathcal{N}$. We know that $(\mathcal{M} \cap \mathcal{N})^{[A]}$ is a subset of $\mathcal{M}^{[A]}$ and since $\mathcal{M}$ is saturated relative to A, we conclude that $I \in \mathcal{M}$. Similarly, $(\mathcal{M} \cap \mathcal{N})^{[\mathcal{P} \setminus A]}$ is a subset of $\mathcal{N}^{[\mathcal{P} \setminus A]}$ and since $\mathcal{N}$ is saturated relative to $\mathcal{P} \setminus A$, we conclude that $I \in \mathcal{N}$. Consequently, $I \in \mathcal{M} \cap \mathcal{N}$. □

Proposition A.29. *Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ be such that $\mathcal{M}$ is semi-saturated relative to A. Then the following equivalences hold:*

(1) $\mathcal{M} = \mathcal{N}$ if and only if $\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{M}^{[A]} \supseteq \mathcal{N}^{[A]}$ and $\mathcal{M}^{[\mathcal{P} \setminus A]} \supseteq \mathcal{N}^{[\mathcal{P} \setminus A]}$.

(2) $\mathcal{M} \subsetneq \mathcal{N}$ if and only if $\mathcal{M} \subseteq \mathcal{N}$ and either $\mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]}$ or $\mathcal{M}^{[\mathcal{P} \setminus A]} \subsetneq \mathcal{N}^{[\mathcal{P} \setminus A]}$.

Proof.

(1) The direct implication follows from Proposition A.15(1). To prove the converse implication, suppose that $\mathcal{M}^{[A]} \supseteq \mathcal{N}^{[A]}$, $\mathcal{M}^{[\mathcal{P} \setminus A]} \supseteq \mathcal{N}^{[\mathcal{P} \setminus A]}$ and $I \in \mathcal{N}$. We immediately obtain $I^{[A]} \in \mathcal{M}^{[A]}$ and $I^{[\mathcal{P} \setminus A]} \in \mathcal{M}^{[\mathcal{P} \setminus A]}$ and since $\mathcal{M}$ is semi-saturated relative to A, we can conclude that $I \in \mathcal{M}$.

(2) For the direct implication suppose that $\mathcal{M} \subsetneq \mathcal{N}$. Then there is some $I \in \mathcal{N}$ such that $I \notin \mathcal{M}$. Since $\mathcal{M}$ is semi-saturated relative to A, we obtain that $I^{[A]} \notin \mathcal{M}^{[A]}$ or $I^{[\mathcal{P} \setminus A]} \notin \mathcal{M}^{[\mathcal{P} \setminus A]}$. Consequently, either $\mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]}$ or $\mathcal{M}^{[\mathcal{P} \setminus A]} \subsetneq \mathcal{N}^{[\mathcal{P} \setminus A]}$. The converse implication is a consequence of Proposition A.15(3). □

Proposition A.30. *Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ such that $\mathcal{N}$ is semi-saturated relative to A. Then the following holds:*

(1) $\mathcal{M} \subseteq \mathcal{N}$ if and only if $\mathcal{M}^{[A]} \subseteq \mathcal{N}^{[A]}$ and $\mathcal{M}^{[\mathcal{P} \setminus A]} \subseteq \mathcal{N}^{[\mathcal{P} \setminus A]}$.

(2) $\mathcal{M} = \mathcal{N}$ if and only if $\mathcal{M}^{[A]} = \mathcal{N}^{[A]}$ and $\mathcal{M}^{[\mathcal{P} \setminus A]} = \mathcal{N}^{[\mathcal{P} \setminus A]}$.

(3) $\mathcal{M} \subsetneq \mathcal{N}$ if and only if $\mathcal{M}^{[A]} \subseteq \mathcal{N}^{[A]}$ and $\mathcal{M}^{[\mathcal{P} \setminus A]} \subseteq \mathcal{N}^{[\mathcal{P} \setminus A]}$ and at least one of the inclusions is proper.

Proof.

(1) The direct implication follows from Proposition A.15(1). To prove the converse implication, suppose that $\mathcal{M}^{[A]} \subseteq \mathcal{N}^{[A]}$, $\mathcal{M}^{[\mathcal{P} \setminus A]} \subseteq \mathcal{N}^{[\mathcal{P} \setminus A]}$ and $I \in \mathcal{M}$. We immediately obtain $I^{[A]} \in \mathcal{M}^{[A]}$, hence also $I^{[A]} \in \mathcal{N}^{[A]}$. Similarly, $I^{[\mathcal{P} \setminus A]} \in \mathcal{M}^{[\mathcal{P} \setminus A]}$ and, consequently, $I^{[\mathcal{P} \setminus A]} \in \mathcal{N}^{[\mathcal{P} \setminus A]}$. Since $\mathcal{N}$ is semi-saturated relative to A, we can conclude that $I \in \mathcal{N}$. Consequently, $\mathcal{M} \subseteq \mathcal{N}$.

(2) This is a consequence of (1).

(3) This is a consequence of (1) and (2). □

Proposition A.31. *Let A be a set of predicate symbols and $\mathcal{M}_1$, $\mathcal{M}_2$ be MKNF interpretations. Then there exists the greatest MKNF interpretation $\mathcal{N}$ that coincides with $\mathcal{M}_1$ on A and with $\mathcal{M}_2$ on $\mathcal{P} \setminus A$. Furthermore, $\mathcal{N}$ is semi-saturated relative to A and $\mathcal{M}_1 \cap \mathcal{M}_2 \subseteq \mathcal{N}$.*

Proof. Let $\mathcal{N}_1 = \sigma(\mathcal{M}_1, A)$, $\mathcal{N}_2 = \sigma(\mathcal{M}_2, \mathcal{P} \setminus A)$ and $\mathcal{N} = \mathcal{N}_1 \cap \mathcal{N}_2$. The claim now follows by Propositions A.22 and A.28. □

A.2.6 Sequence-saturated MKNF Interpretations

The property satisfied by semi-saturated interpretations can be naturally extended to sequences of mutually disjoint sets of predicate symbols. This serves as a means to prove the splitting sequence theorems in Appendix C.

Definition A.32 (Saturation Sequence, Difference Sequence)**.** A *saturation sequence* is a sequence $\langle A_\alpha \rangle_{\alpha<\mu}$ of pairwise disjoint sets of predicate symbols such that $\bigcup_{\alpha<\mu} A_\alpha = \mathcal{P}$.

Definition A.33 (Sequence-saturated MKNF Interpretation)**.** Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence and $\mathcal{M} \in \mathfrak{M}$. We say that $\mathcal{M}$ is *sequence-saturated relative to* $\boldsymbol{A}$ if for every interpretation $I \in \mathfrak{I}$,

$$\forall \alpha < \mu : I^{[A_\alpha]} \in \mathcal{M}^{[A_\alpha]} \qquad \text{implies} \qquad I \in \mathcal{M} \ .$$

Proposition A.34. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence and $\mathcal{M} \in \mathfrak{M}$. Then the following conditions are equivalent:*

1. *$\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{A}$;*
2. *$\mathcal{M} = \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha)$.*
3. *$\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha$ and for any $\alpha < \mu$, $\mathcal{M}_\alpha$ is saturated relative to A_α.*

Proof. We first prove that 1. implies 2. Suppose that $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{A}$. It follows from Proposition A.22 that $\mathcal{M}$ is a subset of $\sigma(\mathcal{M}, A_\alpha)$ for any $\alpha < \mu$. Thus, $\mathcal{M}$ is a subset of $\bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha)$. To show that the converse inclusion holds as well, take some interpretation $I \in \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha)$ and some $\alpha < \mu$. Proposition A.22 implies that $\sigma(\mathcal{M}, A_\alpha)$ coincides with $\mathcal{M}$ on A_α. Thus, from $I \in \sigma(\mathcal{M}, A_\alpha)$ it follows that $I^{[A_\alpha]} \in \mathcal{M}^{[A_\alpha]}$. Furthermore, since $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{A}$, this implies that $I \in \mathcal{M}$.

The implication from 2. to 3. follows by putting $\mathcal{M}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ and observing that, by Proposition A.22, $\mathcal{M}_\alpha$ is saturated relative to A_α.

Finally, suppose that 3. holds and I is an interpretation such that for any $\alpha < \mu$, $I^{[A_\alpha]}$ belongs to

$$\mathcal{M}^{[A_\alpha]} = \left(\bigcap_{\beta<\mu} \mathcal{M}_\beta\right)^{[A_\alpha]} \subseteq \mathcal{M}_\alpha^{[A_\alpha]}$$

Since $\mathcal{M}_\alpha$ is saturated relative to A_α, we conclude that I belongs to $\mathcal{M}_\alpha$. The choice of $\alpha < \mu$ was arbitrary, so we have proven that I belongs to $\mathcal{M}$. □

Proposition A.35. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence, $\langle \mathcal{M}_\alpha \rangle_{\alpha<\mu}$ be a sequence of members of $\mathfrak{M}$ such that for all $\alpha < \mu$, $\mathcal{M}_\alpha$ is saturated relative to A_α, and $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha$. The following holds:*

(1) *$\mathcal{M} = \emptyset$ if and only if for some $\alpha < \mu$, $\mathcal{M}_\alpha = \emptyset$.*

(2) *If $\mathcal{M} \neq \emptyset$, then for all $\alpha < \mu$, $\mathcal{M}_\alpha = \sigma(\mathcal{M}, A_\alpha)$.*

Proof.

(1) Suppose first that $\mathcal{M} \neq \emptyset$. Since $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha$, it follows immediately that for all $\alpha < \mu$, $\mathcal{M}_\alpha \neq \emptyset$.
For the other direction suppose that for all $\alpha < \mu$, $\mathcal{M}_\alpha \neq \emptyset$, pick some $I_\alpha \in \mathcal{M}_\alpha$ for every α and let I be an interpretation such that for every ground atom p,

$$I \models p \qquad \text{if and only if} \qquad \exists \alpha < \mu : I_\alpha^{[A_\alpha]} \models p$$

It follows that for every $\alpha < \mu$, $I^{[A_\alpha]} = I_\alpha^{[A_\alpha]} \in \mathcal{M}_\alpha^{[A_\alpha]}$, so since $\mathcal{M}_\alpha$ is saturated relative to A_α, it follows that $I \in \mathcal{M}_\alpha$. Consequently, $I \in \mathcal{M} \neq \emptyset$.

(2) Suppose that $\mathcal{M} \neq \emptyset$. It follows from (1) that for all $\alpha < \mu$, $\mathcal{M}_\alpha \neq \emptyset$. Pick some $\alpha < \mu$ and some $I \in \mathcal{M}_\alpha$. We will prove that I belongs to $\sigma(\mathcal{M}, A_\alpha)$. Let $I_\alpha = I$

and for all $\beta < \mu$ such that $\beta \neq \alpha$, let I_β be any member of $\mathcal{M}_\beta$. Let J be an interpretation such that for every ground atom p,

$$J \models p \qquad \text{if and only if} \qquad \exists \beta < \mu : I_\beta^{[A_\beta]} \models p \ .$$

To see that J belongs to $\mathcal{M}$, take some $\beta < \mu$ and observe that $J^{[A_\beta]} = I_\beta^{[A_\beta]} \in \mathcal{M}_\beta^{[A_\beta]}$. Since $\mathcal{M}_\beta$ is saturated relative to A_β, this implies that J belongs to $\mathcal{M}_\beta$. Hence, J belongs to $\mathcal{M}$. Moreover, $I^{[A_\alpha]} = J^{[A_\alpha]}$, so $I^{[A_\alpha]}$ belongs to $\mathcal{M}^{[A_\alpha]}$. Thus, I belongs to $\sigma(\mathcal{M}, A_\alpha)$.
For the converse inclusion, suppose that $I \in \sigma(\mathcal{M}, A_\alpha)$ for some $\alpha < \mu$. It follows that $I^{[A_\alpha]} \in \mathcal{M}^{[A_\alpha]}$. Also,

$$\mathcal{M}^{[A_\alpha]} = \left(\bigcap\nolimits_{\beta<\mu} \mathcal{M}_\beta\right)^{[A_\alpha]} \subseteq \mathcal{M}_\alpha^{[A_\alpha]} \ ,$$

so $I^{[A_\alpha]} \in \mathcal{M}_\alpha^{[A_\alpha]}$. Since $\mathcal{M}_\alpha$ is saturated relative to A_α, we obtain $I \in \mathcal{M}_\alpha$. □

A.3 Belief Updates

Proposition 2.24. *Let $\diamond$ be a belief update operator that satisfies (B4.1). Then $\diamond$ satisfies (B8) if and only if it satisfies both (B8.1) and (B8.2).*

Proof. Suppose that $\diamond$ satisfies (B8). It immediately follows that $\diamond$ satisfies (B8.1) because it is a direct weakening of (B8). To see that $\diamond$ satisfies (B8.2), suppose that $\phi \models \psi$. It follows by basic properties of propositional logic that $\phi \diamond \mu \models (\phi \diamond \mu) \vee (\psi \diamond \mu)$ and by (B8) we conclude that $\phi \diamond \mu \models (\phi \vee \psi) \diamond \mu$. Furthermore, from $\phi \models \psi$ we obtain $\phi \vee \psi \equiv \psi$ and, by (B4.1), $(\phi \vee \psi) \diamond \mu \equiv \psi \diamond \mu$. Consequently, $\phi \diamond \mu \models \psi \diamond \mu$.

For the other direction, suppose that $\diamond$ satisfies both (B8.1) and (B8.2). One direction of (B8) follows directly from (B8.1), so we only need to verify that for all propositional formulae ϕ, ψ, μ,

$$(\phi \diamond \mu) \vee (\psi \diamond \mu) \models (\phi \vee \psi) \diamond \mu \ .$$

Note that $\phi \models \phi \vee \psi$ and $\psi \models \phi \vee \psi$, so it follows from (B8.2) that $\phi \diamond \mu \models (\phi \vee \psi) \diamond \mu$ and $\psi \diamond \mu \models (\phi \vee \psi) \diamond \mu$, which establishes our claim. □

A.4 Updates of First-Order Theories

Definition A.36 (Winslett's Operator on Models)**.** Let I be an interpretation and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. We define the operator $\blacklozenge_{\mathsf{W}}$ as follows:

$$I \blacklozenge_{\mathsf{W}} \mathcal{N} = \min\left(\mathcal{N}, \leq_{\mathsf{W}}^{I}\right) \ ,$$
$$\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N} = \bigcup_{I \in \mathcal{M}} I \blacklozenge_{\mathsf{W}} \mathcal{N} = \bigcup_{I \in \mathcal{M}} \min\left(\mathcal{N}, \leq_{\mathsf{W}}^{I}\right) \ .$$

Remark A.37. Note that the above definition is compatible with the definition of Winslett's operator $\diamond_{\mathsf{W}}$ defined in Section 2.6, which operates on first-order theories, in the following sense: for all first-order theories $\mathcal{T}, \mathcal{U}$ it holds that $[\![\mathcal{T} \diamond_{\mathsf{W}} \mathcal{U}]\!] = [\![\mathcal{T}]\!] \blacklozenge_{\mathsf{W}} [\![\mathcal{U}]\!]$.

Lemma A.38. *Suppose that we do not allow for the equality predicate $\approx$. Let A be a set of predicate symbols and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ be both saturated relative to A. Then $\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N}$ is also saturated relative to A.*

Proof. Suppose that J is such that $J^{[A]} \in (\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N})^{[A]}$ but $J \notin (\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N})$. Then there exists some interpretation $J' \in (\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N})$ such that $J'^{[A]} = J^{[A]}$. This also implies that $J' \in \mathcal{N}$ and since $\mathcal{N}$ is saturated relative to A, we obtain that $J \in \mathcal{N}$. Furthermore, there exists some interpretation $I \in \mathcal{M}$ such that $J' \in (I \blacklozenge_{\mathsf{W}} \mathcal{N})$. Let I' be an interpretation such that for every ground atom p,

$$I' \models p \qquad \text{if and only if} \qquad I^{[A]} \models p \lor J^{[\mathcal{P} \setminus A]} \models p \enspace .$$

Then $I'^{[A]} = I^{[A]}$ and $I'^{[\mathcal{P} \setminus A]} = J^{[\mathcal{P} \setminus A]}$ and since $\mathcal{M}$ is saturated relative to A, $I' \in \mathcal{M}$. Since $J \notin (\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N})$, there must exist some interpretation $J'' \in \mathcal{N}$ such that $J'' <_{I'} J$. This means that for every predicate symbol $P \in A$,

$$P^{J''} \div P^{I'} \subseteq P^{J} \div P^{I'} \enspace . \tag{A.8}$$

and for every predicate symbol $P \in \mathcal{P} \setminus A$,

$$P^{J''} \div P^{I'} \subseteq P^{J} \div P^{I'} = \emptyset$$

because I' coincides with J on $\mathcal{P} \setminus A$. Also, for some predicate symbol P_0,

$$P_0^{J''} \div P_0^{I'} \subsetneq P_0^{J} \div P_0^{I'} \enspace .$$

Since this is impossible if P_0 belongs to $\mathcal{P} \setminus A$, P_0 must belong to A. Let J''' be an interpretation such that for every ground atom p,

$$J''' \models p \qquad \text{if and only if} \qquad J''^{[A]} \models p \lor J'^{[\mathcal{P} \setminus A]} \models p \enspace .$$

By (A.8), for predicate symbols $P \in A$ it holds that

$$P^{J'''} \div P^{I} = P^{J''} \div P^{I'} \subseteq P^{J} \div P^{I'} = P^{J'} \div P^{I}$$

and for predicate symbols $P \in \mathcal{P} \setminus A$ we obtain

$$P^{J'''} \div P^{I} = P^{J'} \div P^{I} \enspace .$$

Also, for P_0 it holds that $P_0^{J'''} \div P_0^{I} = P_0^{J''} \div P_0^{I'} \subsetneq P_0^{J} \div P_0^{I'} = P_0^{J'} \div P_0^{I}$. As a consequence, $J''' <^{I}_{\mathsf{W}} J'$. Furthermore, since $J'''^{[A]} = J''^{[A]} \in \mathcal{N}^{[A]}$, it follows that $J''' \in \mathcal{N}$, so we arrived at a conflict with the assumption that $J' \in (I \blacklozenge_{\mathsf{W}} \mathcal{N})$. □

Theorem 2.43. *If we do not allow for the equality predicate $\approx$, then Winslett's first-order update operator $\diamond_{\mathsf{W}}$ conserves the language.*

Proof. Follows by induction on i from Proposition 2.41 and Lemma A.38. □

Theorem 2.45. *Winslett's first-order update operator $\diamond_{\mathsf{W}}$ respects fact update.*

Proof. Let $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$ be a finite sequence of consistent sets of ground objective literals. We prove by induction on n:

1° If $n = 1$, then $[\![\diamond_{\mathsf{W}} \boldsymbol{T}]\!] = [\![\mathcal{T}_0]\!] = [\![\{ l \in \mathcal{L}_{\mathsf{G}} \mid l \in \mathcal{T}_0 \}]\!]$, which establishes the claim.

2° Let $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$, $\mathcal{M} = [\![\diamond_{\mathsf{W}} \boldsymbol{T}]\!]$ and $\boldsymbol{T}' = \langle \mathcal{T}_i \rangle_{i<n+1}$. It follows that $(\diamond_{\mathsf{W}} \boldsymbol{T}') = (\diamond_{\mathsf{W}} \boldsymbol{T}) \diamond_{\mathsf{W}} \mathcal{T}_n$ and by the definition of $\diamond_{\mathsf{W}}$,

$$\mathcal{M}' = [\![\diamond_{\mathsf{W}} \boldsymbol{T}']\!] = \bigcup_{I \in \mathcal{M}} \min \left([\![\mathcal{T}_n]\!], \leq_{\mathsf{W}}^{I} \right) \quad . \tag{A.9}$$

By the inductive assumption for $\boldsymbol{T}$ we obtain that $I \in \mathcal{M}$ if and only if

$$I \models \left\{ l \in \mathcal{L}_{\mathsf{G}} \,\middle|\, \exists j < n : l \in \mathcal{T}_j \land (\forall i : j < i < n \Longrightarrow \bar{l} \notin \mathcal{T}_i) \right\} \quad . \tag{A.10}$$

Our goal is to prove that $I' \in \mathcal{M}'$ if and only if

$$I' \models \left\{ l \in \mathcal{L}_{\mathsf{G}} \,\middle|\, \exists j < n+1 : l \in \mathcal{T}_j \land (\forall i : j < i < n+1 \Longrightarrow \bar{l} \notin \mathcal{T}_i) \right\} \quad . \tag{A.11}$$

Take some $I' \in \mathcal{M}'$. Then it follows from (A.9) that there is some $I \in \mathcal{M}$ such that $I' \in \min \left([\![\mathcal{T}_n]\!], \leq_{\mathsf{W}}^{I} \right)$. Hence, $I' \models \{ l \in \mathcal{L}_{\mathsf{G}} \mid l \in \mathcal{T}_n \}$ and by the definition of $\leq_{\mathsf{W}}^{I}$ we can conclude that I' and I can differ only in the interpretation of ground atoms p such that either $p \in \mathcal{T}_n$ or $\neg p \in \mathcal{T}_n$. Consequently, since I satisfies (A.10), we conclude that I' satisfies (A.11).
For the converse inclusion, suppose that I' satisfies (A.11) and let I be an interpretation that satisfies (A.10) and intereprets a minimal set of ground atoms differently from I'. By (A.11) we obtain that $I' \models \mathcal{T}_n$ and I can differ from I' only in the interpretation of ground atoms p such that either $p \in \mathcal{T}_n$ or $\neg p \in \mathcal{T}_n$. Thus, we can conclude that $I' \in \min \left([\![\mathcal{T}_n]\!], \leq_{\mathsf{W}}^{I} \right)$. Furthermore, since I satisfies (A.10), it follows that $I \in \mathcal{M}$. This implies that $I' \in \mathcal{M}'$. □

Proposition 2.46. *Every first-order update operator that is characterised by a faithful preorder assignment satisfies postulates (FO1) – (FO6) and (FO8.2).*

Proof. Let $\diamond$ be a first-order update operator characterised by a faithful preorder assignment ω. Then, for all first-order theories $\mathcal{T}, \mathcal{U}$,

$$[\![\mathcal{T} \diamond \mathcal{U}]\!] = \bigcup_{I \in [\![\mathcal{T}]\!]} \min \left([\![\mathcal{U}]\!], \leq_{\omega}^{I} \right) \quad . \tag{A.12}$$

We consider each main postulate separately; postulates (FO2.⊤), (FO2.1), (FO2.2), (FO4.1) and (FO4.2) are their direct consequences:

(FO1) It follows from (A.12) that all interpretations from $[\![\mathcal{T} \diamond \mathcal{U}]\!]$ are members of $[\![\mathcal{U}]\!]$. In other words, $\mathcal{T} \diamond \mathcal{U} \models \mathcal{U}$.

(FO2) Suppose that $\mathcal{T} \models \mathcal{U}$ and take some $I \in [\![\mathcal{T}]\!] \subseteq [\![\mathcal{U}]\!]$. Since the preorder assignment is faithful, for all $J \in [\![\mathcal{U}]\!]$ with $J \neq I$ we have $I <_{\omega}^{I} J$. Consequently, $\min([\![\mathcal{U}]\!], \leq_{\omega}^{I}) = \{ I \}$ and from (A.12) it follows that $\mathcal{T} \diamond \mathcal{U} \equiv \mathcal{T}$.

(FO3) Suppose that both $\mathcal{T}$ and $\mathcal{U}$ are satisfiable. Then there is some $I_0 \in [\![\mathcal{T}]\!]$ and also some $J_0 \in \min([\![\mathcal{U}]\!], \leq_{\omega}^{I_0})$, so we obtain $J_0 \in [\![\mathcal{T} \diamond \mathcal{U}]\!]$. Hence $\mathcal{T} \diamond \mathcal{U}$ is satisfiable.

(FO4) If $\mathcal{T} \equiv \mathcal{S}$ and $\mathcal{U} \equiv \mathcal{V}$, then

$$[\![\mathcal{T} \diamond \mathcal{U}]\!] = \bigcup_{I \in [\![\mathcal{T}]\!]} \min \left([\![\mathcal{U}]\!], \leq_{\omega}^{I} \right) = \bigcup_{I \in [\![\mathcal{S}]\!]} \min \left([\![\mathcal{V}]\!], \leq_{\omega}^{I} \right) = [\![\mathcal{S} \diamond \mathcal{V}]\!] \quad .$$

Therefore, $\mathcal{T} \diamond \mathcal{U} \equiv \mathcal{S} \diamond \mathcal{V}$.

(FO5) Suppose that J is a model of $(\mathcal{T} \diamond \mathcal{U}) \cup \mathcal{V}$. Then $J \in [\![\mathcal{V}]\!]$ and there is some model I of $\mathcal{T}$ such that J belongs to $\min([\![\mathcal{U}]\!], \leq_\omega^I)$. Consequently, J also belongs to $\min([\![\mathcal{U}]\!] \cap [\![\mathcal{V}]\!], \leq_\omega^I)$, so J is a model of $\mathcal{T} \diamond (\mathcal{U} \cup \mathcal{V})$.

(FO6) Assume that $\mathcal{T} \diamond \mathcal{U} \models \mathcal{V}$ and $\mathcal{T} \diamond \mathcal{V} \models \mathcal{U}$. We will prove by contradiction that $\mathcal{T} \diamond \mathcal{U} \models \mathcal{T} \diamond \mathcal{V}$. The other half can be proved similarly.
So suppose that J is a model of $\mathcal{T} \diamond \mathcal{U}$ but not of $\mathcal{T} \diamond \mathcal{V}$. Then there is some model I of $\mathcal{T}$ such that

$$J \in \min\left([\![\mathcal{U}]\!], \leq_\omega^I\right) \quad . \tag{A.13}$$

At the same time, there must be some model K of $\mathcal{V}$ such that $K <_\omega^I J$. Let K_0 be minimal w.r.t. $\leq_\omega^I$ among all such K. Then by transitivity of $<_\omega^I$ we obtain that $K_0 \in \min\left([\![\mathcal{V}]\!], \leq_\omega^I\right)$ and, consequently, K_0 is a model of $\mathcal{T} \diamond \mathcal{V}$. By the assumption we now obtain that K_0 is a model of $\mathcal{U}$. But since $K_0 <_\omega^I J$, this is in conflict with (A.13).

(FO8.2) Suppose that $\mathcal{T} \models \mathcal{S}$. It immediately follows from (A.12) that $[\![\mathcal{T} \diamond \mathcal{U}]\!]$ is a subset of $[\![\mathcal{S} \diamond \mathcal{U}]\!]$. Thus $\mathcal{T} \diamond \mathcal{U} \models \mathcal{S} \diamond \mathcal{U}$. □

A.5 Rule Updates

Theorem 2.70 (Respect for Support and Language Conservation). *Let* X *be one of* D, W, B *and* $i \in \{0, 1, 2\}$. *The rule update semantics* AS, JU, DS, RD, PRZ, PRX$_i$, RVS *and* RVD *respect support and conserve the language.*

Proof (sketch). Note that language conservation is a direct consequence of support, so it suffices to prove support. The arguments presented in what follows frequently rely on the following three facts:

1. All stable models of a logic program $\mathcal{P}$ are supported by $\mathcal{P}$.
2. If an interpretation is supported by a program $\mathcal{P}$, then it is also supported by every superset of $\mathcal{P}$.
3. If an interpretation is supported by the expansion $\mathcal{P}^e$ of $\mathcal{P}$, then it is also supported by $\mathcal{P}$ (because the additional rules in the expansion have default literals in their heads).

The claim for the AS- and JU-semantics follows directly from the fact that AS- and JU-models are stable models of a subset of $\mathsf{all}(\boldsymbol{P}^e)$, so they must be supported by $\mathsf{all}(\boldsymbol{P})$.

The case of the DS- and RD-semantics is only slightly more involved than the previous one. Any DS- or RD-model J of a DLP $\boldsymbol{P}$ satisfies

$$J' = \mathsf{least}(\mathcal{Q} \cup \mathsf{def}(\boldsymbol{P}^e, J)) \quad ,$$

for some subset $\mathcal{Q}$ of $\mathsf{all}(\boldsymbol{P}^e)$. The definition of $\mathsf{least}(\cdot)$ and J' allows us to conclude that every objective literal from J must be supported by $\mathcal{Q} \cup \mathsf{def}(\boldsymbol{P}, J)$. Furthermore, $\mathsf{def}(\boldsymbol{P}, J)$ contains only rules with default literals in their heads, so every objective literal from J is supported $\mathcal{Q}$, and thus also by $\mathsf{all}(\boldsymbol{P})$.

In case of the PRZ-semantics, the claim follows from the fact that every result of an update of a program $\mathcal{P}$ by a program $\mathcal{U}$ is a reduct of the form $\mathcal{P}' \cup \mathcal{U}$ where $\mathcal{P}'$ is a subset of $\mathcal{P}$. So every PRZ-model of $\langle \mathcal{P}, \mathcal{U} \rangle$ is a stable model of $\mathcal{P}' \cup \mathcal{U}$, thus it is supported by $\mathcal{P} \cup \mathcal{U} = \mathsf{all}(\langle \mathcal{P}, \mathcal{U} \rangle)$. Essentially the same argument works for the RVS-semantics as well.

Suppose that X is one of D, W, B and $i \in \{0, 1, 2\}$. It follows that every PRX$_i$-model J of a DLP $\boldsymbol{P}$ is also a stable model of some subset of $\mathsf{all}(\boldsymbol{P}) \cup \mathsf{all}(\boldsymbol{P})^d$. Thus it is supported by $\mathsf{all}(\boldsymbol{P}) \cup \mathsf{all}(\boldsymbol{P})^d$. Moreover, whenever a literal is supported by a rule π^d, it is also supported by the original rule π. Consequently, J is supported by $\mathsf{all}(\boldsymbol{P})$.

Finally, given a DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, the construction of an RVD-model ensures that every objective literal added to it at a stage i is supported by some rule from the program $\mathcal{P}_{n-i}$. Furthermore, both objective and default components of RVD-models monotonically grow during their construction, so support for a literal can never be retracted in further stages (for details please consult (Delgrande, 2010)). □

Theorem 2.72 (Respect for Fact Update). *Let X be one of* D*,* W*,* B *and* $i \in \{0, 1, 2\}$*. The rule update semantics* AS*,* JU*,* DS*,* RD*,* PRZ*,* PRX$_i$*,* RVS *and* RVD *respect fact update.*

Proof (sketch). Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a sequence of consistent sets of facts.

First consider the JU-semantics. When sets of facts are considered, the set of rejected rules is independent of the model candidate J. Furthermore, the following set of unrejected rules remains:

$$\{\, l. \mid \exists j < n : (l.) \in \mathcal{P}_j \wedge (\forall i : j < i < n \Longrightarrow (\sim l.) \notin \mathcal{P}_i^{\mathsf{e}}) \,\} \\ \cup \left\{ \sim l. \;\middle|\; \exists j < n : (\sim l.) \in \mathcal{P}_j^{\mathsf{e}} \wedge (\forall i : j < i < n \Longrightarrow (l.) \notin \mathcal{P}_i) \right\} \quad \text{(A.14)}$$

Since $\mathcal{P}_j$ is consistent for all $j < n$, this set must also be consistent (all inconsistencies across different components of $\boldsymbol{P}$ have been resolved). Thus $\boldsymbol{P}$ has the desired JU-model

$$\left\{ l \;\middle|\; \exists j < n : (l.) \in \mathcal{P}_j \wedge (\forall i : j < i < n \Longrightarrow \{\, \bar{l}., \sim l. \,\} \cap \mathcal{P}_i = \emptyset) \right\} \;.$$

Virtually the same arguments apply to the DS- and RD-semantics and according to (Homola, 2004), the AS-semantics coincides with the JU-semantics on $\boldsymbol{P}$ because $\mathsf{all}(\boldsymbol{P})$ is acyclic.

Now consider the PRZ-semantics and a DLP $\langle \mathcal{P}, \mathcal{U} \rangle$ where $\mathcal{P}, \mathcal{U}$ are consistent sets of facts. Following the process of updating $\mathcal{P}$ by $\mathcal{U}$, we first take the unique stable model $J_{\mathcal{P}}$ of $\mathcal{P}$ and update it by $\mathcal{U}$, obtaining the new interpretation

$$J_{\langle \mathcal{P}, \mathcal{U} \rangle} = \left\{ l \in \mathcal{L}_{\mathsf{G}} \;\middle|\; (l.) \in \mathcal{U} \vee ((l.) \in \mathcal{P} \wedge \{\, \bar{l}., \sim l. \,\} \cap \mathcal{U} = \emptyset) \right\}$$

It follows that $\mathcal{P}' \cup \mathcal{U}$, where $\mathcal{P}'$ is a maximal subset of $\mathcal{P}$ coherent with $J_{\langle \mathcal{P}, \mathcal{U} \rangle}$, has the desired stable model.

Turning to the PRX$_i$-semantics, a case-by-case analysis of the preference relations D, W and B reveals that regardless of the particular operator $*_0$, $*_1$ or $*_2$, they only allow for a single preferred stable model which coincides with the desired one. We leave the verification to the reader; the precise definitions of the preference strategies can be found in (Schaub and Wang, 2003; Delgrande et al., 2007).

In case of the RVS-semantics, the facts from $\mathcal{P}$ that are inconsistent with $\mathcal{U}$ must be eliminated and, by the maximality condition, no other facts can be eliminated. Thus the desired model is the model of $\mathcal{P}' \cup \mathcal{U}$ where $\mathcal{P}'$ is the greatest subset of $\mathcal{P}$ that is coherent with $\mathcal{U}$.

Finally, given sequence of consistent sets of facts $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, the RVD-models are constructed in stages from $\mathcal{P}_{n-1}$ backwards towards $\mathcal{P}_0$, always keeping the greatest set of facts that is consistent with previously adopted ones. This directly leads to the desired model. For more details the reader can refer to (Delgrande, 2010). □

Theorem 2.74 (Respect for Causal Rejection). *The rule update semantics* AS*,* JU*,* DS *and* RD *respect causal rejection.*

Proof (sketch). Take some AS-, JU-, DS- or RD-model J of a DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$. If $J \not\models \pi$ for some $\pi \in \mathcal{P}_i$, then it follows from the definition of these semantics that π must be a rejected rule. In case of the AS-, JU- and DS-semantics, the claim then follows immediately by the definition of the set of rejected rules.

As for the RD-semantics, the same argument applies except when π is rejected by some $\sigma \in \mathcal{P}_i$ and no more preferred rule rejects π. But in this case it follows that all rules from $\mathsf{all}(\boldsymbol{P}^{\mathsf{e}})$ with the head $H(\pi)$ or $H(\sigma) = {\sim}H(\pi)$ are also rejected. As a consequence, the fixpoint condition of the RD-semantics cannot be satisfied, so J cannot be an RD-model of $\boldsymbol{P}$ – a contradiction with the assumption. □

Theorem 2.76 (Respect for Acyclic Justified Update). *The rule update semantics AS, JU, DS and RD respect acyclic justified update.*

Proof (sketch). The case of the JU-semantics is obvious and proofs for AS- and DS-semantics can be found in (Homola, 2004). Analogous arguments apply to the RD-semantics as well. □

B

Proofs: ABox Updates for Hybrid MKNF Knowledge Bases

In the following we present proofs of results from Chapter 3, implicitly working under the same assumptions as those imposed in that chapter. That is, we assume that all MKNF rules are ground and non-disjunctive and do not allow for rules with empty heads and for rules with default negation in their heads. In the proofs we also implicitly use Propositions A.1 and A.2 as well as Corollaries A.10 and A.11.

B.1 Static Consequence Operator

Lemma B.1 (Monotonicity of $T_{\mathcal{P}}$)**.** *Let $\mathcal{P}$ be a positive MKNF program and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. If $\mathcal{M} \subseteq \mathcal{N}$, then $T_{\mathcal{P}}(\mathcal{M}) \supseteq T_{\mathcal{P}}(\mathcal{N})$.*

Proof. Suppose that the generalised atom ξ belongs to $T_{\mathcal{P}}(\mathcal{N})$. Then there is a rule $\pi \in \mathcal{P}$ such that $\mathcal{N} \models \kappa(B(\pi))$ and $H(\pi) = \{ \xi \}$. It follows from Proposition A.1 that $\mathcal{M} \models \kappa(B(\pi))$ and, thus, $\xi \in T_{\mathcal{P}}(\mathcal{M})$. ☐

Proposition 3.4 (Monotonicity of $T_{\mathcal{K}}$). *Let $\mathcal{K}$ be a positive MKNF knowledge base. Then $T_{\mathcal{K}}$ is a monotonic function on the complete lattice $(\mathfrak{M}, \subseteq)$.*

Proof. Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ and take some $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ such that $\mathcal{M} \subseteq \mathcal{N}$. Our goal is to show that $T_{\mathcal{K}}(\mathcal{M}) \subseteq T_{\mathcal{K}}(\mathcal{N})$.

It follows from Lemma B.1 that $T_{\mathcal{P}}(\mathcal{M}) \supseteq T_{\mathcal{P}}(\mathcal{N})$ and, consequently,

$$T_{\mathcal{K}}(\mathcal{M}) = [\![T_{\mathcal{P}}(\mathcal{M}) \cup \kappa(\mathcal{O})]\!] \subseteq [\![T_{\mathcal{P}}(\mathcal{N}) \cup \kappa(\mathcal{O})]\!] = T_{\mathcal{K}}(\mathcal{N}) \enspace . \qquad ☐$$

Lemma B.2. *Let $\mathcal{K}$ be a positive MKNF knowledge base. An MKNF interpretation $\mathcal{M}$ is an S5 model of $\mathcal{K}$ if and only if $\mathcal{M} \subseteq T_{\mathcal{K}}(\mathcal{M})$.*

Proof. First suppose that $\mathcal{M}$ is an S5 model of $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. Then clearly $\mathcal{M} \models \kappa(\mathcal{O})$ and for every rule $\pi \in \mathcal{P}$ such that $\mathcal{M} \models \kappa(B(\pi))$ it holds that $\mathcal{M} \models H(\pi)$. In other words, $\mathcal{M}$

is an S5 model of $T_{\mathcal{P}}(\mathcal{M}) \cup \kappa(\mathcal{O})$ and since $T_{\mathcal{K}}(\mathcal{M})$ is its greatest S5 model, it follows that $\mathcal{M} \subseteq T_{\mathcal{K}}(\mathcal{M})$.

On the other hand, if $\mathcal{M} \subseteq T_{\mathcal{K}}(\mathcal{M})$, then $\mathcal{M} \models T_{\mathcal{P}}(\mathcal{M}) \cup \kappa(\mathcal{O})$. It follows that $\mathcal{M} \models \mathbf{K}\,\kappa(\phi)$ for every $\phi \in \mathcal{O}$ and whenever $\mathcal{M} \models \kappa(B(\pi))$, it holds that $\mathcal{M} \models \kappa(H(\pi))$. Consequently, $\mathcal{M}$ is an S5 model of $\mathcal{K}$. □

Lemma B.3. *Every positive MKNF knowledge base either has no S5 model, or it has a unique MKNF model which coincides with its greatest S5 model.*

Proof. Suppose that the MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ has some S5 model and let $\mathcal{M}$ be the union of all S5 models of $\mathcal{K}$. First we show that $\mathcal{M}$ is an S5 model of $\mathcal{K}$, i.e. it is the greatest S5 model of $\mathcal{M}$.

Take some MKNF sentence ϕ from $\kappa(\mathcal{K})$. If $\phi = \mathbf{K}\,\kappa(\phi)$ for some $\phi \in \mathcal{O}$, then since $\kappa(\phi)$ is a first-order sentence, it follows from Proposition A.2 that $\mathcal{M} \models \mathbf{K}\,\kappa(\phi)$.

The other possibility is that ϕ is a sentence of the form $\bigwedge \kappa(B(\pi)) \supset \bigvee \kappa(H(\pi))$ for some $\pi \in \mathcal{P}$. Suppose that $\mathcal{M} \models \kappa(B(\pi))$. As π is positive, it follows from Proposition A.1 that $\mathcal{N} \models \kappa(B(\pi))$ for every S5 model $\mathcal{N}$ of $\mathcal{K}$ and, thus, $\mathcal{N} \models \kappa(H(\pi))$. Hence, by Proposition A.2, $\mathcal{M} \models \kappa(H(\pi))$.

It remains to show that $\mathcal{M}$ is the unique MKNF model of $\mathcal{K}$. Since $\kappa(\mathcal{K})$ is subjective **not**-free, it follows by the definitions of MKNF satisfaction and of an MKNF model that the MKNF models of $\mathcal{K}$ are exactly its subset-maximal S5 models. Since $\mathcal{M}$ is the greatest S5 model of $\mathcal{M}$, it follows that it is also its unique MKNF model. □

Proposition 3.5 (MKNF Model of a Positive MKNF Knowledge Base). *Let $\mathcal{K}$ be a positive MKNF knowledge base. An MKNF interpretation is an MKNF model of $\mathcal{K}$ if and only if it is the greatest fixed point of $T_{\mathcal{K}}$.*

Proof. Let $\mathcal{S} = \{\, \mathcal{M} \mid \mathcal{M} \subseteq T_{\mathcal{K}}(\mathcal{M}) \,\}$ and $\mathcal{M}^* = \bigcup \mathcal{S}$. It follows that for every $\mathcal{M} \in \mathcal{S}$, $\mathcal{M} \subseteq \mathcal{M}^*$ and, by Proposition 3.4, we obtain that $\mathcal{M} \subseteq T_{\mathcal{K}}(\mathcal{M}) \subseteq T_{\mathcal{K}}(\mathcal{M}^*)$. Hence,

$$\mathcal{M}^* = \bigcup_{\mathcal{M} \in \mathcal{S}} \mathcal{M} \subseteq T_{\mathcal{K}}(\mathcal{M}^*)$$

and we conclude that $\mathcal{M}^*$ belongs to $\mathcal{S}$. Then it follows by the monotonicity of $T_{\mathcal{K}}$ that $T_{\mathcal{K}}(\mathcal{M}^*)$ belongs to $\mathcal{S}$ and thus $T_{\mathcal{K}}(\mathcal{M}^*) \subseteq \mathcal{M}^*$. Consequently, $\mathcal{M}^*$ is a fixed point of $T_{\mathcal{K}}$. Furthermore, every fixed point of $T_{\mathcal{K}}$ belongs to $\mathcal{S}$, so $\mathcal{M}^*$ is its greatest fixed point.

Now it suffices to observe that, by Lemma B.2, $\mathcal{S}$ consists of all S5 models of $\mathcal{K}$ and the empty set. If $\mathcal{M}^* = \emptyset$, then $\mathcal{K}$ has no S5 model, and thus no MKNF model. On the other hand, if $\mathcal{M}^* \neq \emptyset$, then $\mathcal{M}^*$ is the greatest S5 model of $\mathcal{K}$ and, by Lemma B.3, it coincides with the unique MKNF model of $\mathcal{K}$. □

Lemma B.4. *Let $\mathcal{K}$ be an MKNF knowledge base. An MKNF interpretation $\mathcal{M}$ is an S5 model of $\mathcal{K}$ if and only if $\mathcal{M}$ is an S5 model of $\mathcal{K}^{\mathcal{M}}$.*

Proof. Suppose that $\mathcal{M}$ is an S5 model of $\mathcal{K}$. Obviously, $\mathcal{M} \models \mathbf{K}\,\kappa(\phi)$ for every $\phi \in \mathcal{O}$. Take some $\pi' = (H(\pi) \leftarrow B(\pi)^+.)$ from $\mathcal{P}^{\mathcal{M}}$ for some $\pi \in \mathcal{P}$ with $\mathcal{M} \models \kappa({\sim}B(\pi)^-)$. Then $\kappa(\mathcal{K}^{\mathcal{M}})$ contains the sentence $\kappa(\pi')$ of the form $\bigwedge \kappa(B(\pi)^+) \supset \bigvee \kappa(H(\pi))$. If $\mathcal{M} \models \kappa(B(\pi)^+)$, then it it follows that $\mathcal{M} \models \kappa(B(\pi))$ and since $\mathcal{M} \models \kappa(\pi)$, it follows that $\mathcal{M} \models \kappa(H(\pi))$. Therefore, $\mathcal{M} \models \kappa(\pi')$.

For the converse implication, assume that $\mathcal{M}$ is an S5 model of $\mathcal{K}^{\mathcal{M}}$. Obviously, $\mathcal{M} \models \mathbf{K}\,\kappa(\phi)$ for every $\phi \in \mathcal{O}$, so consider some rule $\pi \in \mathcal{P}$. If $\mathcal{M} \models \kappa(B(\pi))$, then $\mathcal{P}^{\mathcal{M}}$ contains the rule $\pi' = (H(\pi) \leftarrow B(\pi)^+.)$ and since $\mathcal{M} \models \kappa(\pi')$, it follows that $\mathcal{M} \models \kappa(H(\pi))$. Hence, $\mathcal{M} \models \kappa(\pi)$. □

Proposition 3.8 (MKNF Model of an MKNF Knowledge Base). *Let $\mathcal{K}$ be an MKNF knowledge base. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if it is the MKNF model of $\mathcal{K}^{\mathcal{M}}$.*

Proof. Suppose that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$. Then it is also an S5 model of $\mathcal{K}$, so it follows that it is an S5 model of $\mathcal{K}^{\mathcal{M}}$ from Lemma B.4.

Since $\mathcal{M}$ is an S5 model of $\mathcal{K}^{\mathcal{M}}$, it must hold that $\mathcal{M}$ is a subset of the greatest S5 model $\mathcal{M}'$ of $\mathcal{K}^{\mathcal{M}}$. We show by contradiction that $\mathcal{M} = \mathcal{M}'$, i.e. $\mathcal{M}$ is the MKNF model of $\mathcal{K}^{\mathcal{M}}$ (c.f. Lemma B.3).

Assume that $\mathcal{M} \subsetneq \mathcal{M}'$. Since $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, there must be some sentence $\phi \in \kappa(\mathcal{K})$ such that $(\mathcal{M}', \mathcal{M}) \not\models \phi$. But $\mathcal{M}' \models \mathbf{K}\, \kappa(\psi)$ for every $\psi \in \mathcal{O}$, so ϕ must be of the form $\kappa(B(\pi)) \supset \kappa(H(\pi))$ for some rule $\pi \in \mathcal{P}$ and the following must hold:

$$(\mathcal{M}', \mathcal{M}) \models \kappa(\sim B(\pi)^-) \land (\mathcal{M}', \mathcal{M}) \models \kappa(B(\pi)^+) \land (\mathcal{M}', \mathcal{M}) \not\models \kappa(H(\pi))$$

which is equivalent to

$$\mathcal{M} \models \kappa(\sim B(\pi)^-) \land \mathcal{M}' \models \kappa(B(\pi)^+) \land \mathcal{M}' \not\models \kappa(H(\pi)) \ .$$

However, this is in conflict with $\mathcal{M}'$ being an S5 model of $\mathcal{K}^{\mathcal{M}}$ since the sentence

$$\bigwedge \kappa(B(\pi)^+) \supset \bigvee \kappa(H(\pi))$$

belongs to $\kappa(\mathcal{K}^{\mathcal{M}})$.

For the converse implication, assume that $\mathcal{M}$ is the MKNF model of $\mathcal{K}^{\mathcal{M}}$. Then it follows from Lemma B.4 that $\mathcal{M}$ is an S5 model of $\mathcal{K}$.

Take some $\mathcal{M}' \supsetneq \mathcal{M}$. Since $\mathcal{M}$ is the greatest S5 model of $\mathcal{K}^{\mathcal{M}}$, there is some rule $\pi' = (H(\pi) \leftarrow B(\pi)^+.) \in \mathcal{P}^{\mathcal{M}}$ such that $\mathcal{M}' \not\models \kappa(\pi')$, i.e.

$$\mathcal{M} \models \kappa(\sim B(\pi)^-) \land \mathcal{M}' \models \kappa(B(\pi)^+) \land \mathcal{M}' \not\models H(\pi) \ .$$

This is equivalent to

$$(\mathcal{M}', \mathcal{M}) \models \kappa(\sim B(\pi)^-) \land (\mathcal{M}', \mathcal{M}) \models \kappa(B(\pi)^+) \land (\mathcal{M}', \mathcal{M}) \not\models \kappa(H(\pi))$$

which in turn is equivalent to $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\pi)$. This proves that $\mathcal{M}$ is indeed an MKNF model of $\mathcal{K}$. □

B.2 Updating Consequence Operator

Proposition 3.12 (Monotonicity of $T^{\diamond}_{\boldsymbol{K}}$). *Let $\diamond$ be a first-order update operator and $\boldsymbol{K}$ a positive DMKB with static rules. If $\diamond$ satisfies (FO8.2), then $T^{\diamond}_{\boldsymbol{K}}$ is a monotonic function on the complete lattice $(\mathfrak{M}, \subseteq)$.*

Proof. Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$ and take some $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ such that $\mathcal{M} \subseteq \mathcal{N}$. Our goal is to show that $T^{\diamond}_{\boldsymbol{K}}(\mathcal{M}) \subseteq T^{\diamond}_{\boldsymbol{K}}(\mathcal{N})$.

By Lemma B.1 we conclude that $T_{\mathcal{P}_0}(\mathcal{M}) \supseteq T_{\mathcal{P}_0}(\mathcal{N})$. Consequently,

$$T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0) \models T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0) \ .$$

By repeatedly using (FO8.2) for all $\mathcal{O}_i$ with $0 < i < n$ we obtain that

$$((T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0)) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}) \models ((T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0)) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}) \enspace .$$

Consequently, $T^{\diamond}_{\boldsymbol{K}}(\mathcal{M}) \subseteq T^{\diamond}_{\boldsymbol{K}}(\mathcal{N})$. □

B.3 Properties and Relations

Theorem 3.19 (Faithfulness w.r.t. MKNF Knowledge Bases). *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\langle \mathcal{K} \rangle$.*

Proof. This follows from Propositions 3.5 and 3.8, and the fact that for every first-order theory $\mathcal{T}$, $\Diamond\langle \mathcal{T} \rangle = \mathcal{T}$, so the static consequence operator $T_{\mathcal{K}}$ coincides with the updating consequence operator $T^{\diamond}_{\langle \mathcal{K} \rangle}$. □

Theorem 3.20 (Faithfulness w.r.t. First-Order Update Operator). *Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \emptyset) \rangle_{i<n}$ be a DMKB. An MKNF interpretation $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M} = [\![\Diamond\langle \mathcal{O}_i \rangle_{i<n}]\!]$.*

Proof. Since the $\boldsymbol{K}$ contains no rules, $\boldsymbol{K}^{\mathcal{M}} = \boldsymbol{K}$ for any $\mathcal{M} \in \mathfrak{M}$. Furthermore,

$$T^{\diamond}_{\boldsymbol{K}}(\mathcal{M}) = [\![\Diamond\langle \mathcal{O}_i \rangle_{i<n}]\!] \enspace .$$

It follows that the only fixed point of $T^{\diamond}_{\boldsymbol{K}}$ is the set of models $[\![\Diamond\langle \mathcal{O}_i \rangle_{i<n}]\!]$. □

Theorem 3.21 (Primacy of New Information). *Suppose that $\diamond$ satisfies (FO1) and let $\boldsymbol{K} = \langle \mathcal{K}_i \rangle_{i<n}$ be a DMKB with static rules such that $n > 0$. If $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}$, then $\mathcal{M} \models \kappa(\mathcal{K}_{n-1})$.*

Proof. Suppose that $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}$. Then $\mathcal{M}$ is a fixed point of the operator $T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}$, i.e.

$$\mathcal{M} = T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{M}) = \left[\!\!\left[\left(T_{\mathcal{P}_0^{\mathcal{M}}}(\mathcal{M}) \cup \kappa(\mathcal{O}_0) \right) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1} \right]\!\!\right] \enspace .$$

Let

$$\mathcal{T} = \left(T_{\mathcal{P}_0^{\mathcal{M}}}(\mathcal{M}) \cup \kappa(\mathcal{O}_0) \right) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-2} \enspace .$$

It follows that

$$\mathcal{M} = [\![\mathcal{T} \diamond \mathcal{O}_{n-1}]\!]$$

and by (FO1) we can conclude that $\mathcal{M} \models \kappa(\mathcal{O}_{n-1})$. Consequently, $\mathcal{M} \models \kappa(\mathcal{K}_{n-1})$. □

Theorem 3.22 (Immunity to Tautological Updates). *Suppose that $\diamond$ satisfies (FO2.⊤) and (FO4) and let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$ be a DMKB with static rules such that $\mathcal{O}_j \equiv \emptyset$ for some j with $0 < j < n$ and*

$$\boldsymbol{K}' = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n \wedge i \neq j} \enspace .$$

Then $\boldsymbol{K}$ and $\boldsymbol{K}'$ have the same $\diamond$-dynamic MKNF models.

Proof. Let $\mathcal{M}$ be an MKNF interpretation and $\mathcal{N} \in \mathfrak{M}$. We will prove that $T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N})$, which implies that the $\diamond$-dynamic MKNF models of $\boldsymbol{K}$ and $\boldsymbol{K}'$ coincide. Put

$$\mathcal{T} = \left(T_{\mathcal{P}_0^{\mathcal{M}}}(\mathcal{N}) \cup \kappa(\mathcal{O}_0)\right) \diamond \mathcal{O}_1 \diamond \ldots \diamond \mathcal{O}_{j-1} \enspace .$$

It follows that

$$T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = [\![\mathcal{T} \diamond \mathcal{O}_j \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \quad \text{and} \quad T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N}) = [\![\mathcal{T} \diamond \mathcal{O}_{j+1} \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \enspace .$$

Let $\mathcal{T}' = \mathcal{T} \diamond \mathcal{O}_j$. By (FO4) and (FO2.$\top$) we obtain that $\mathcal{T}' \equiv \mathcal{T}$ and by repeated application of (FO4) we conclude that

$$\mathcal{T}' \diamond \mathcal{O}_{j+1} \diamond \cdots \diamond \mathcal{O}_{n-1} \equiv \mathcal{T} \diamond \mathcal{O}_{j+1} \cdots \diamond \mathcal{O}_{n-1} \enspace .$$

Consequently, $T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N})$. □

Theorem 3.23 (Syntax Independence). *Suppose that $\diamond$ satisfies (FO4). Let $\boldsymbol{K} = \langle(\mathcal{O}_i, \mathcal{P}_i)\rangle_{i<n}$ and $\boldsymbol{K}' = \langle(\mathcal{O}'_i, \mathcal{P}'_i)\rangle_{i<n}$ be DMKBs with static rules such that $\mathcal{P}_0 = \mathcal{P}'_0$ and $\mathcal{O}_i \equiv \mathcal{O}'_i$ for all $i < n$. Then $\boldsymbol{K}$ and $\boldsymbol{K}'$ have the same $\diamond$-dynamic MKNF models.*

Proof. Let $\mathcal{M}$ be an MKNF interpretation and $\mathcal{N} \in \mathfrak{M}$. We will prove that $T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N})$, which implies that the $\diamond$-dynamic MKNF models of $\boldsymbol{K}$ and $\boldsymbol{K}'$ coincide. Observe that

$$T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = [\![(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0)) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \enspace ,$$
$$T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N}) = [\![\left(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}'_0)\right) \diamond \mathcal{O}'_1 \diamond \cdots \diamond \mathcal{O}'_{n-1}]\!] \enspace .$$

It follows from $\mathcal{O}_0 \equiv \mathcal{O}'_0$ that

$$T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0) \equiv T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}'_0)$$

and by applying (FO4) we obtain that

$$(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0)) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1} \equiv \left(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}'_0)\right) \diamond \mathcal{O}'_1 \cdots \diamond \mathcal{O}'_{n-1} \enspace .$$

Consequently, $T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N})$. □

C

Proofs: Layered Dynamic MKNF Knowledge Bases

In the following we present proofs of results from Chapter 4, implicitly working under the same assumptions as those imposed in that chapter. That is, we constrain ourselves to a generalised atom base that consists of objective literals, meaning that MKNF programs coincide with logic programs and we assume that every rule is ground and has exactly one literal in its head. Furthermore, we do not consider the equality predicate because it interferes with language conservation of Winslett's first-order update operator (c.f. Example 2.35).

C.1 Semantics with Splitting Properties

C.1.1 MKNF Knowledge Bases

Remark C.1. Note that whenever U is a splitting set for an MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, the following holds:

$$\mathsf{pr}(b_U(\mathcal{O})) \subseteq U \enspace, \qquad \mathsf{pr}(b_U(\mathcal{P})) \subseteq U \enspace, \qquad \mathsf{pr}(b_U(\mathcal{K})) \subseteq U \enspace, \qquad \mathsf{pr}(t_U(\mathcal{O})) \subseteq \mathcal{P} \setminus U \enspace.$$

Also note that the heads of rules in $t_U(\mathcal{P})$ contain only predicate symbols from $\mathcal{P} \setminus U$ while their bodies may also contain predicate symbols from U. However, for any $\mathcal{X} \in \mathcal{M}$, the reducts $e_U(\mathcal{P}, \mathcal{X})$ and $e_U(\mathcal{K}, \mathcal{X})$ contain only predicate symbols from $\mathcal{P} \setminus U$:

$$\mathsf{pr}(e_U(\mathcal{P}, \mathcal{X})) \subseteq \mathcal{P} \setminus U \enspace, \qquad \qquad \mathsf{pr}(e_U(\mathcal{K}, \mathcal{X})) \subseteq \mathcal{P} \setminus U \enspace.$$

These basic observations will be used in the following proofs without further notice or reference.

C.1.1.1 Splitting Set Theorem

Definition C.2 (Generalised Splitting Set Reduct)**.** Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base, $U \subseteq \mathbb{P}$ a set of predicate symbols and $\mathcal{X}, \mathcal{X}' \in \mathcal{M}$. We define the *reduct of $\mathcal{P}$ relative to U and $(\mathcal{X}', \mathcal{X})$* as

$$e_U(\mathcal{P}, (\mathcal{X}', \mathcal{X})) = \{\, H(\pi) \leftarrow \{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq \mathbb{P} \setminus U \,\}. \mid \pi \in t_U(\mathcal{P}) \\ \wedge (\mathcal{X}', \mathcal{X}) \models \kappa(\{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq U \,\}) \,\} \enspace .$$

The *reduct of $\mathcal{K}$ relative to U and $(\mathcal{X}', \mathcal{X})$* is $e_U(\mathcal{K}, (\mathcal{X}', \mathcal{X})) = (t_U(\mathcal{O}), e_U(\mathcal{P}, (\mathcal{X}', \mathcal{X})))$.

Lemma C.3. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{D}$, $\mathcal{D}'$, $\mathcal{E}$, $\mathcal{E}'$, $\mathcal{F}$, $\mathcal{F}'$, $\mathcal{G}$, $\mathcal{G}' \in \mathcal{M}$ be such that the following conditions are satisfied:*

1. $\mathcal{E}^{[U]} = \mathcal{D}^{[U]}$ *and* $\mathcal{E}'^{[U]} = \mathcal{D}'^{[U]}$*;*
2. $\mathcal{F}^{[\mathbb{P} \setminus U]} = \mathcal{D}^{[\mathbb{P} \setminus U]}$ *and* $\mathcal{F}'^{[\mathbb{P} \setminus U]} = \mathcal{D}'^{[\mathbb{P} \setminus U]}$*;*
3. $\mathcal{G}^{[U]} = \mathcal{D}^{[U]}$ *and* $\mathcal{G}'^{[U]} = \mathcal{D}'^{[U]}$*.*

Then,

$$(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K}) \qquad \textit{implies} \qquad (\mathcal{E}', \mathcal{E}) \models \kappa(b_U(\mathcal{K})) \wedge (\mathcal{F}', \mathcal{F}) \models \kappa(e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))) \enspace .$$

Proof. Suppose that $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K})$ and let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. Since $b_U(\mathcal{O}) \subseteq \mathcal{O}$, it follows that for every $\phi \in b_U(\mathcal{O})$, $(\mathcal{D}', \mathcal{D}) \models \mathbf{K}\, \kappa(\phi)$. Also, every rule $\pi \in b_U(\mathcal{P})$ belongs to $\mathcal{P}$ and we can conclude that $(\mathcal{D}', \mathcal{D}) \models \kappa(\pi)$. Consequently, $(\mathcal{D}', \mathcal{D}) \models \kappa(b_U(\mathcal{K}))$ and from Corollary A.17 we obtain $(\mathcal{E}', \mathcal{E}) \models \kappa(b_U(\mathcal{K}))$.

It remains to show that $(\mathcal{F}', \mathcal{F}) \models \kappa(e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G})))$. We know that $e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G})) = (t_U(\mathcal{O}), e_U(\mathcal{P}, (\mathcal{G}', \mathcal{G})))$. Since $t_U(\mathcal{O}) \subseteq \mathcal{O}$, it follows that for every $\phi \in t_U(\mathcal{O})$, $(\mathcal{D}', \mathcal{D}) \models \mathbf{K}\, \kappa(\phi)$ and we can use Corollary A.17 to conclude that $(\mathcal{F}', \mathcal{F}) \models \mathbf{K}\, \kappa(t_U(\phi))$.

Take some rule $\sigma \in e_U(\mathcal{P}, (\mathcal{G}', \mathcal{G}))$. By Corollary A.9, in order to prove that $(\mathcal{F}', \mathcal{F}) \models \kappa(\sigma)$, we can instead show that $(\mathcal{F}', \mathcal{F}) \models \kappa(H(\sigma))$ given the assumption that $(\mathcal{F}', \mathcal{F}) \models \kappa(B(\sigma))$. This assumption together with Corollary A.17 implies that $(\mathcal{D}', \mathcal{D}) \models \kappa(B(\sigma))$ and, by the definition of $e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))$, there must be some rule $\pi \in \mathcal{P}$ such that $H(\pi) = H(\sigma)$ and $(\mathcal{G}', \mathcal{G}) \models \kappa(B(\pi) \setminus B(\sigma))$. From the last property and Corollary A.17 we obtain $(\mathcal{D}', \mathcal{D}) \models \kappa(B(\pi) \setminus B(\sigma))$. Thus, $(\mathcal{D}', \mathcal{D}) \models \kappa(B(\pi))$ and since $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K})$ and $\kappa(\mathcal{K})$ contains $\kappa(\pi)$, we conclude that $(\mathcal{D}', \mathcal{D}) \models \kappa(H(\pi))$. Consequently, since $H(\pi) = H(\sigma)$, by Corollary A.17 we obtain $(\mathcal{F}', \mathcal{F}) \models \kappa(H(\sigma))$ and so $(\mathcal{F}', \mathcal{F}) \models \kappa(\sigma)$. The choice of σ was arbitrary, so we have proven that $(\mathcal{F}', \mathcal{F}) \models \kappa(e_U(\mathcal{P}, (\mathcal{G}', \mathcal{G})))$. □

Lemma C.4. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{D}$, $\mathcal{D}'$, $\mathcal{E}$, $\mathcal{E}'$, $\mathcal{F}$, $\mathcal{F}'$, $\mathcal{G}$, $\mathcal{G}' \in \mathcal{M}$ be such that the following conditions are satisfied:*

1. $\mathcal{E}^{[U]} = \mathcal{D}^{[U]}$ *and* $\mathcal{E}'^{[U]} = \mathcal{D}'^{[U]}$*;*
2. $\mathcal{F}^{[\mathbb{P} \setminus U]} = \mathcal{D}^{[\mathbb{P} \setminus U]}$ *and* $\mathcal{F}'^{[\mathbb{P} \setminus U]} = \mathcal{D}'^{[\mathbb{P} \setminus U]}$*;*
3. $\mathcal{G}^{[U]} = \mathcal{D}^{[U]}$ *and* $\mathcal{G}'^{[U]} = \mathcal{D}'^{[U]}$*.*

Then,

$$(\mathcal{E}', \mathcal{E}) \models \kappa(b_U(\mathcal{K})) \wedge (\mathcal{F}', \mathcal{F}) \models \kappa(e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))) \qquad \textit{implies} \qquad (\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K}) \enspace .$$

Proof. Take some $\phi \in \kappa(\mathcal{K})$ and let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. If $\phi = \mathbf{K}\, \kappa(\psi)$ for some $\psi \in \mathcal{O}$, then using Corollary A.17 we can conclude that from $\mathcal{O} = b_U(\mathcal{O}) \cup t_U(\mathcal{O})$ and our assumptions it follows that $(\mathcal{D}', \mathcal{D}) \models \mathbf{K}\, \kappa(\phi)$.

Now suppose that $\phi = \kappa(\pi)$ for some $\pi \in \mathcal{P}$. If $\pi \in b_U(\mathcal{P})$, then it follows that $(\mathcal{E}', \mathcal{E}) \models \kappa(\pi)$ and by Corollary A.17 we conclude that $(\mathcal{D}', \mathcal{D}) \models \kappa(\pi)$. If $\pi \in t_U(\mathcal{P})$, then assuming that that $(\mathcal{D}', \mathcal{D}) \models \kappa(B(\pi))$ we need to prove that $(\mathcal{D}', \mathcal{D}) \models \kappa(H(\pi))$. It follows by Corollary A.17 that

$$(\mathcal{G}', \mathcal{G}) \models \kappa(\{ L \in B(\pi) \mid \mathsf{pr}(L) \subseteq U \})$$

Consequently, $e_U(\mathcal{P}, (\mathcal{G}', \mathcal{G}))$ contains a rule σ such that $H(\sigma) = H(\pi)$ and

$$B(\sigma) = \{ L \in B(\pi) \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus U \} \ .$$

Thus, by Corollary A.17, $(\mathcal{F}', \mathcal{F}) \models \kappa(B(\sigma))$ and from $(\mathcal{F}', \mathcal{F}) \models e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))$ it follows that $(\mathcal{F}', \mathcal{F}) \models \kappa(H(\sigma))$. By another application of Corollary A.17 we obtain that $(\mathcal{D}', \mathcal{D}) \models \kappa(H(\sigma))$ and since $H(\sigma) = H(\pi)$, it follows that $(\mathcal{D}', \mathcal{D}) \models \kappa(\pi)$. Consequently, $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K})$. ☐

Proposition C.5. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{D}$, $\mathcal{D}'$, $\mathcal{E}$, $\mathcal{E}'$, $\mathcal{F}$, $\mathcal{F}'$, $\mathcal{G}$, $\mathcal{G}' \in \mathfrak{M}$ be such that the following conditions are satisfied:*

1. $\mathcal{E}^{[U]} = \mathcal{D}^{[U]}$ *and* $\mathcal{E}'^{[U]} = \mathcal{D}'^{[U]}$;
2. $\mathcal{F}^{[\mathcal{P} \setminus U]} = \mathcal{D}^{[\mathcal{P} \setminus U]}$ *and* $\mathcal{F}'^{[\mathcal{P} \setminus U]} = \mathcal{D}'^{[\mathcal{P} \setminus U]}$;
3. $\mathcal{G}^{[U]} = \mathcal{D}^{[U]}$ *and* $\mathcal{G}'^{[U]} = \mathcal{D}'^{[U]}$.

Then,

$$(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K}) \quad \textit{if and only if} \quad (\mathcal{E}', \mathcal{E}) \models \kappa(b_U(\mathcal{K})) \land (\mathcal{F}', \mathcal{F}) \models \kappa(e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))) \ .$$

Proof. Follows by Lemmas C.3 and C.4. ☐

Corollary C.6. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M}$, $\mathcal{X}$ be MKNF interpretations such that $\mathcal{X}^{[U]} = \mathcal{M}^{[U]}$. Then,*

$$\mathcal{M} \models \kappa(\mathcal{K}) \qquad \textit{if and only if} \qquad \mathcal{M} \models \kappa(b_U(\mathcal{K})) \land \mathcal{M} \models \kappa(e_U(\mathcal{K}, \mathcal{X})) \ .$$

Proof. Proposition C.5 for $\mathcal{D} = \mathcal{D}' = \mathcal{E} = \mathcal{E}' = \mathcal{F} = \mathcal{F}' = \mathcal{M}$ and $\mathcal{G} = \mathcal{G}' = \mathcal{X}$ implies that

$$(\mathcal{M}, \mathcal{M}) \models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{M}, \mathcal{M}) \models \kappa(b_U(\mathcal{K})) \land (\mathcal{M}, \mathcal{M}) \models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

The claim now follows from Proposition A.8(2). ☐

Corollary C.7. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M}$, $\mathcal{X}$, $\mathcal{Y}$ be MKNF interpretations such that $\mathcal{X}^{[U]} = \mathcal{M}^{[U]}$ and $\mathcal{Y}^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]}$. Then,*

$$\mathcal{M} \models \kappa(\mathcal{K}) \qquad \textit{if and only if} \qquad \mathcal{X} \models \kappa(b_U(\mathcal{K})) \land \mathcal{Y} \models \kappa(e_U(\mathcal{K}, \mathcal{X})) \ .$$

Proof. Proposition C.5 for $\mathcal{D} = \mathcal{D}' = \mathcal{M}$, $\mathcal{E} = \mathcal{E}' = \mathcal{X}$, $\mathcal{F} = \mathcal{F}' = \mathcal{Y}$ and $\mathcal{G} = \mathcal{G}' = \mathcal{X}$ implies that

$$(\mathcal{M}, \mathcal{M}) \models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{X}, \mathcal{X}) \models \kappa(b_U(\mathcal{K})) \land (\mathcal{Y}, \mathcal{Y}) \models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

The claim now follows from Proposition A.8(2). ☐

Corollary C.8. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M}$, $\mathcal{M}'$, $\mathcal{X}$, $\mathcal{X}'$ be MKNF interpretations such that $\mathcal{M} \models \kappa(\mathcal{K})$, $\mathcal{M}'^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]}$, $\mathcal{X}^{[U]} = \mathcal{M}^{[U]}$ and $\mathcal{X}'^{[U]} =$*

$\mathcal{M}'^{[U]}$. *Then,*

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \qquad \textit{implies} \qquad (\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \ .$$

Proof. Proposition C.5 for $\mathcal{D} = \mathcal{M}$, $\mathcal{D}' = \mathcal{M}'$, $\mathcal{E} = \mathcal{X}$, $\mathcal{E}' = \mathcal{X}'$, $\mathcal{F} = \mathcal{F}' = \mathcal{M}$, $\mathcal{G} = \mathcal{G}' = \mathcal{X}$ implies that

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \lor (\mathcal{M}, \mathcal{M}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

Furthermore, from Corollary C.6 we know that $\mathcal{M} \models \kappa(e_U(\mathcal{K}, \mathcal{X}))$ is satisfied because $\mathcal{M} \models \kappa(\mathcal{K})$. Hence, by Proposition A.8(2) the second disjunct on the right hand side of the above equivalence can be safely omitted and we obtain the claim of this corollary. □

Corollary C.9. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M}$, $\mathcal{M}'$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Y}'$ be MKNF interpretations such that $\mathcal{M} \models \kappa(\mathcal{K})$, $\mathcal{M}'^{[U]} = \mathcal{X}^{[U]} = \mathcal{M}^{[U]}$, $\mathcal{Y}^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]}$ and $\mathcal{Y}'^{[\mathcal{P} \setminus U]} = \mathcal{M}'^{[\mathcal{P} \setminus U]}$. Then,*

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \qquad \textit{implies} \qquad (\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X})) \ .$$

Proof. Proposition C.5 for $\mathcal{D} = \mathcal{M}$, $\mathcal{D}' = \mathcal{M}'$, $\mathcal{E} = \mathcal{E}' = \mathcal{M}$, $\mathcal{F} = \mathcal{Y}$, $\mathcal{F}' = \mathcal{Y}'$, $\mathcal{G} = \mathcal{G}' = \mathcal{X}$ implies that

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{M}, \mathcal{M}) \not\models \kappa(b_U(\mathcal{K})) \lor (\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

Furthermore, from Corollary C.6 we know that $\mathcal{M} \models \kappa(b_U(\mathcal{K}))$ is satisfied because $\mathcal{M} \models \kappa(\mathcal{K})$. Hence, by Proposition A.8(2), the first disjunct on the right hand side of the above equivalence can be safely omitted and we obtain the claim of this corollary. □

Corollary C.10. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M}$, $\mathcal{M}'$, $\mathcal{X}$, $\mathcal{X}'$ be MKNF interpretations such that $\mathcal{X}^{[U]} = \mathcal{M}^{[U]}$ and $\mathcal{X}'^{[U]} = \mathcal{M}'^{[U]}$. Then,*

$$(\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \qquad \textit{implies} \qquad (\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \ .$$

Proof. Proposition C.5 for $\mathcal{D} = \mathcal{M}$, $\mathcal{D}' = \mathcal{M}'$, $\mathcal{E} = \mathcal{X}$, $\mathcal{E}' = \mathcal{X}'$, $\mathcal{F} = \mathcal{G} = \mathcal{M}$, $\mathcal{F}' = \mathcal{G}' = \mathcal{M}'$ implies that

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \lor (\mathcal{M}', \mathcal{M}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{M}', \mathcal{M}))) \ .$$

The claim of this corollary follows directly from this equivalence. □

Corollary C.11. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M}$, $\mathcal{M}'$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Y}'$ be MKNF interpretations such that $\mathcal{M}'^{[U]} = \mathcal{X}^{[U]} = \mathcal{M}^{[U]}$, $\mathcal{Y}^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]}$ and $\mathcal{Y}'^{[\mathcal{P} \setminus U]} = \mathcal{M}'^{[\mathcal{P} \setminus U]}$. Then,*

$$(\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X})) \qquad \textit{implies} \qquad (\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \ .$$

Proof. Proposition C.5 for $\mathcal{D} = \mathcal{M}$, $\mathcal{D}' = \mathcal{M}'$, $\mathcal{E} = \mathcal{E}' = \mathcal{M}$, $\mathcal{F} = \mathcal{Y}$, $\mathcal{F}' = \mathcal{Y}'$, $\mathcal{G} = \mathcal{G}' = \mathcal{X}$ implies that

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{M}, \mathcal{M}) \not\models \kappa(b_U(\mathcal{K})) \lor (\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

The claim of this corollary follows from this equivalence by Proposition A.8(2). □

Proposition C.12. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$, $\mathcal{M}$ an MKNF model of $\mathcal{K}$ and $\mathcal{X} = \sigma(\mathcal{M}, U)$. Then $\mathcal{X}$ is an MKNF model of $b_U(\mathcal{K})$.*

Proof. By Proposition A.22 we know that $\mathcal{M} \subseteq \mathcal{X}$ and that $\mathcal{X}$ is saturated relative to U. By Corollary A.11, $\mathcal{X}$ is an MKNF model of $b_U(\mathcal{K})$ if and only if $\mathcal{X} \models \kappa(b_U(\mathcal{K}))$ and for every $\mathcal{X}' \supsetneq \mathcal{X}$ it holds that $(\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K}))$. The former follows directly from Corollary C.7. To verify the latter, pick some $\mathcal{X}' \supsetneq \mathcal{X}$. By Proposition A.31 there exists the greatest MKNF interpretation $\mathcal{M}'$ that coincides with $\mathcal{X}'$ on U (i.e. $\mathcal{M}'^{[U]} = \mathcal{X}'^{[U]}$) and with $\mathcal{M}$ on $\mathcal{P} \setminus U$ (i.e. $\mathcal{M}'^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]}$) and which includes $\mathcal{X}' \cap \mathcal{M}$. Hence,

$$\mathcal{M} \subseteq \mathcal{X} \cap \mathcal{M} \subseteq \mathcal{X}' \cap \mathcal{M} \subseteq \mathcal{M}' \ . \tag{C.1}$$

Furthermore, we know that $\mathcal{X}$ is saturated relative to U, so we can use Proposition A.19(2) to conclude that

$$\mathcal{M}^{[U]} = \mathcal{X}^{[U]} \subsetneq \mathcal{X}'^{[U]} = \mathcal{M}'^{[U]} \ . \tag{C.2}$$

Consequently, by (C.1), (C.2) and Proposition A.15(3), we obtain $\mathcal{M} \subsetneq \mathcal{M}'$. This, together with the assumption that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, implies that $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$. We can now apply Corollary C.8 to conclude that $(\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K}))$, which is the desired conclusion. □

Proposition C.13. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$, $\mathcal{M}$ an MKNF model of $\mathcal{K}$, $\mathcal{X} = \sigma(\mathcal{M}, U)$ and $\mathcal{Y} = \sigma(\mathcal{M}, \mathcal{P} \setminus U)$. Then $\mathcal{Y}$ is an MKNF model of $e_U(\mathcal{K}, \mathcal{X})$.*

Proof. By Proposition A.22 we know that $\mathcal{M} \subseteq \mathcal{Y}$ and that $\mathcal{Y}$ is saturated relative to $\mathcal{P} \setminus U$. By Corollary A.11, $\mathcal{Y}$ is an MKNF model of $e_U(\mathcal{K}, \mathcal{X})$ if and only if $\mathcal{Y} \models \kappa(e_U(\mathcal{K}, \mathcal{X}))$ and for every $\mathcal{Y}' \supsetneq \mathcal{Y}$ it holds that $(\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X}))$. The former follows directly from Corollary C.7. To verify the latter, pick some $\mathcal{Y}' \supsetneq \mathcal{Y}$. By Proposition A.31 there exists the greatest MKNF interpretation $\mathcal{M}'$ that coincides with $\mathcal{M}$ on U (i.e. $\mathcal{M}'^{[U]} = \mathcal{M}^{[U]}$) and with $\mathcal{Y}'$ on $\mathcal{P} \setminus U$ (i.e. $\mathcal{M}'^{[\mathcal{P} \setminus U]} = \mathcal{Y}'^{[\mathcal{P} \setminus U]}$) and which includes $\mathcal{M} \cap \mathcal{Y}'$. Hence,

$$\mathcal{M} \subseteq \mathcal{M} \cap \mathcal{Y} \subseteq \mathcal{M} \cap \mathcal{Y}' \subseteq \mathcal{M}' \ . \tag{C.3}$$

Furthermore, we know that $\mathcal{Y}$ is saturated relative to $\mathcal{P} \setminus U$, so we can use Proposition A.19(2) to conclude that

$$\mathcal{M}^{[\mathcal{P} \setminus U]} = \mathcal{Y}^{[\mathcal{P} \setminus U]} \subsetneq \mathcal{Y}'^{[\mathcal{P} \setminus U]} = \mathcal{M}'^{[\mathcal{P} \setminus U]} \ . \tag{C.4}$$

Consequently, by (C.3), (C.4) and Proposition A.15(3), we obtain $\mathcal{M} \subsetneq \mathcal{M}'$. This, together with the assumption that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, implies that $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$. We can now apply Corollary C.9 to conclude that $(\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X}))$, which is the desired conclusion. □

Proposition C.14. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $(\mathcal{X}, \mathcal{Y})$ a solution to $\mathcal{K}$ w.r.t. U. Then $\mathcal{X} \cap \mathcal{Y}$ is an MKNF model of $\mathcal{K}$.*

Proof. Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ and $\mathcal{M} = \mathcal{X} \cap \mathcal{Y}$. In order to show that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, we need to prove that $\mathcal{M} \models \kappa(\mathcal{K})$ and that for every $\mathcal{M}' \supsetneq \mathcal{M}$ it holds that $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$. We verify the two conditions separately.

We know that $\mathcal{X}$ is an MKNF model of $b_U(\mathcal{K})$, so, by Proposition A.18, $\mathcal{X}$ is saturated relative to U. Similarly, since $\mathcal{Y}$ is an MKNF model of $e_U(\mathcal{K}, \mathcal{X})$, it must be saturated relative to $\mathcal{P} \setminus U$. Hence, by Proposition A.28, $\mathcal{M}$ is semi-saturated relative to U, $\mathcal{M}^{[U]} = \mathcal{X}^{[U]}$ and $\mathcal{M}^{[\mathcal{P} \setminus U]} = \mathcal{Y}^{[\mathcal{P} \setminus U]}$.

Since $(\mathcal{X}, \mathcal{Y})$ is a solution to $\mathcal{K}$ w.r.t. U, $\mathcal{X}$ must an MKNF model of $b_U(\mathcal{K})$ and $\mathcal{Y}$ an MKNF model of $e_U(\mathcal{K}, \mathcal{X})$. So $\mathcal{X} \models \kappa(b_U(\mathcal{K}))$ and $\mathcal{Y} \models \kappa(e_U(\mathcal{K}, \mathcal{X}))$. Consequently, by Corollary C.7, $\mathcal{M} \models \kappa(\mathcal{K})$.

Now take some MKNF interpretation $\mathcal{M}' \supsetneq \mathcal{M}$ and let $\mathcal{X}' = \mathcal{X} \cup \mathcal{M}'$ and $\mathcal{Y}' = \mathcal{Y} \cup \mathcal{M}'$. We already inferred that $\mathcal{M}$ is semi-saturated relative to U, which means that by Proposition A.29(2) one of the following cases must occur:

a) If $\mathcal{M}'^{[U]} \supsetneq \mathcal{M}^{[U]}$, then

$$\mathcal{X}'^{[U]} = \mathcal{X}^{[U]} \cup \mathcal{M}'^{[U]} = \mathcal{M}'^{[U]} \supsetneq \mathcal{M}^{[U]} = \mathcal{X}^{[U]} \enspace ,$$

so Proposition A.15(3) implies that $\mathcal{X}' \supsetneq \mathcal{X}$. Hence, since $\mathcal{X}$ is an MKNF model of $b_U(\mathcal{K})$, we infer that $(\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K}))$ and by Corollary C.10 we obtain $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$ as desired.

b) If $\mathcal{M}'^{[U]} = \mathcal{M}^{[U]}$ and $\mathcal{M}'^{[\mathcal{P} \setminus U]} \supsetneq \mathcal{M}^{[\mathcal{P} \setminus U]}$, then $\mathcal{X}'^{[U]} = \mathcal{M}'^{[U]} = \mathcal{M}^{[U]}$ and

$$\mathcal{Y}'^{[\mathcal{P} \setminus U]} = \mathcal{Y}^{[\mathcal{P} \setminus U]} \cup \mathcal{M}'^{[\mathcal{P} \setminus U]} = \mathcal{M}'^{[\mathcal{P} \setminus U]} \supsetneq \mathcal{M}^{[\mathcal{P} \setminus U]} = \mathcal{Y}^{[\mathcal{P} \setminus U]} \enspace ,$$

so Proposition A.15(3) implies that $\mathcal{Y}' \supsetneq \mathcal{Y}$. Hence, since $\mathcal{Y}$ is an MKNF model of $e_U(\mathcal{K}, \mathcal{X})$, we infer that $(\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X}))$ and by Corollary C.11 we obtain $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$ as desired. □

Theorem C.15 (Splitting Set Theorem for MKNF Knowledge Bases). *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M} = \mathcal{X} \cap \mathcal{Y}$ for some solution $(\mathcal{X}, \mathcal{Y})$ to $\mathcal{K}$ w.r.t. U.*

Proof. First suppose that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$. By Proposition C.12 we obtain that $\mathcal{X} = \sigma(\mathcal{M}, U)$ is an MKNF model of $b_U(\mathcal{K})$ and by Proposition C.13 that $\mathcal{Y} = \sigma(\mathcal{M}, \mathcal{P} \setminus U)$ is an MKNF model of $e_U(\mathcal{K}, \mathcal{X})$. Furthermore, by Proposition A.22, $\mathcal{X}^{[U]} = \mathcal{M}^{[U]}$ and $\mathcal{X}$ is saturated relative to U, $\mathcal{Y}^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]}$ and $\mathcal{Y}$ is saturated relative to $\mathcal{P} \setminus U$, and $\mathcal{M} \subseteq \mathcal{X} \cap \mathcal{Y}$. To prove the converse inclusion, note that the following holds by Proposition A.28:

$$(\mathcal{X} \cap \mathcal{Y})^{[U]} = \mathcal{X}^{[U]} = \mathcal{M}^{[U]} \enspace ,$$
$$(\mathcal{X} \cap \mathcal{Y})^{[\mathcal{P} \setminus U]} = \mathcal{Y}^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]} \enspace .$$

Suppose that $\mathcal{M} \subsetneq \mathcal{X} \cap \mathcal{Y}$. Then, since $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, $(\mathcal{X} \cap \mathcal{Y}, \mathcal{M}) \not\models \kappa(\mathcal{K})$ and by Proposition C.5 for $\mathcal{D} = \mathcal{M}$, $\mathcal{D}' = \mathcal{X} \cap \mathcal{Y}$, $\mathcal{E} = \mathcal{E}' = \mathcal{F} = \mathcal{F}' = \mathcal{M}$, $\mathcal{G} = \mathcal{G}' = \mathcal{X}$, we obtain

$$(\mathcal{M}, \mathcal{M}) \not\models \kappa(b_U(\mathcal{K})) \lor (\mathcal{M}, \mathcal{M}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \enspace .$$

However, Proposition A.8(2) and Corollary C.6 now entail that $\mathcal{M} \not\models \kappa(\mathcal{K})$, a conflict with the assumption that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$. Consequently, $\mathcal{M} = \mathcal{X} \cap \mathcal{Y}$.

The converse implication follows directly from Proposition C.14. □

Corollary C.16. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M} \in \mathbb{M}$. If $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, then the pair $(\sigma(\mathcal{M}, U), \sigma(\mathcal{M}, \mathcal{P} \setminus U))$ is a solution to $\mathcal{K}$ w.r.t. U, $\mathcal{M} = \sigma(\mathcal{M}, U) \cap \sigma(\mathcal{M}, \mathcal{P} \setminus U)$ and $\mathcal{M}$ is semi-saturated relative to U.*

Proof. This is a consequence of the proof of Theorem C.15 and of Proposition A.28. □

Corollary C.17. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ such that there exists at least one solution to $\mathcal{K}$ w.r.t. U. Then $\mathcal{K}$ is MKNF satisfiable and an MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M} = \mathcal{X} \cap \mathcal{Y}$ for some solution $(\mathcal{X}, \mathcal{Y})$ to $\mathcal{K}$ w.r.t. U.*

Proof. Follows from Theorem C.15. □

Corollary C.18. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$. An MKNF sentence ϕ is MKNF entailed by $\mathcal{K}$ if and only if for every solution $(\mathcal{X}, \mathcal{Y})$ of $\mathcal{K}$ w.r.t. U, $\mathcal{X} \cap \mathcal{Y} \models \phi$.*

Proof. Follows from Theorem C.15. □

C.1.1.2 Splitting Sequence Theorem

Remark C.19. Solutions to an MKNF knowledge base $\mathcal{K}$ w.r.t. a splitting sequence $(U, \mathcal{P})$ are the same as the solutions to $\mathcal{K}$ w.r.t. the splitting set U.

Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$, and let $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ be a sequence of MKNF interpretations. Then,

$$\mathsf{pr}(b_{U_0}(\mathcal{K})) \subseteq U_0 \ ,$$
$$\mathsf{pr}\left(e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right)\right) \subseteq U_{\alpha+1} \setminus U_\alpha \text{ whenever } \alpha+1<\mu$$

Furthermore, if $\boldsymbol{X}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$, then $\mathcal{X}_0$ is saturated relative to U_0 and for every α such that $\alpha+1<\mu$, $\mathcal{X}_{\alpha+1}$ is saturated relative to $U_{\alpha+1} \setminus U_\alpha$. Also note that for any limit ordinal α, $\mathcal{X}_\alpha = \mathcal{I}$, so $\mathcal{X}_\alpha$ is saturated relative to any set of predicate symbols.

The proofs in this section follow the same pattern as those in (Turner, 1996).

Lemma C.20. *Let $\langle U_\alpha \rangle_{\alpha<\mu}$ be a sequence of sets of predicate symbols and $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ be a sequence of members of $\mathcal{M}$ such that for all $\alpha < \mu$, $\mathcal{X}_\alpha$ is saturated relative to U_α. Then $\bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ is saturated relative to $\bigcup_{\alpha<\mu} U_\alpha$.*

Proof. Let $U = \bigcup_{\alpha<\mu} U_\alpha$ and $\mathcal{X} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ and suppose that $I^{[U]}$ belongs to $\mathcal{X}^{[U]}$. Then there is some $J \in \mathcal{X}$ such that $I^{[U]} = J^{[U]}$. This means that for every ground atom p with $\mathsf{pr}(p) \subseteq U$,

$$I \models p \qquad \text{if and only if} \qquad J \models p \ .$$

We need to show that I belongs to $\mathcal{X}$. Take some $\beta < \mu$ and some atom q such that $\mathsf{pr}(q) \subseteq U_\beta$. Since U_β is a subset of U, we obtain

$$I \models q \qquad \text{if and only if} \qquad J \models q \ .$$

It follows that $I^{[U_\beta]} = J^{[U_\beta]}$ and since $J \in \mathcal{X} \subseteq \mathcal{X}_\beta$, we conclude that $I^{[U_\beta]} \in \mathcal{X}_\beta^{[U_\beta]}$. Moreover, $\mathcal{X}_\beta$ is saturated relative to U_β, so $I \in \mathcal{X}_\beta$. Since the choice of β was arbitrary, I belongs to $\mathcal{X}_\beta$ for all $\beta < \mu$. Thus, $I \in \mathcal{X}$ as desired. □

Definition C.21 (Saturation Sequence Induced by a Splitting Sequence)**.** Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a monotone, continuous sequence of sets of predicate symbols. The *saturation sequence induced by* $\boldsymbol{U}$ is the sequence $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ where

- $A_0 = U_0$;
- for any ordinal α such that $\alpha+1<\mu$, $A_{\alpha+1} = U_{\alpha+1} \setminus U_\alpha$;
- for any limit ordinal α, $A_\alpha = \emptyset$.

Lemma C.22. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$, $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ the saturation sequence induced by $\boldsymbol{U}$ and $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ a sequence of members of $\mathcal{M}$ such that $\mathcal{X}_\alpha$ is saturated relative to A_α for every $\alpha < \mu$. Then the following holds for all ordinals α, β such that $\beta < \alpha < \mu$:*

- $\bigcap_{\gamma \leq \beta} \mathcal{X}_\gamma$ *is saturated relative to* U_β*;*
- $\bigcap_{\beta < \gamma \leq \alpha} \mathcal{X}_\gamma$ *is saturated relative to* $U_\alpha \setminus U_\beta$*;*
- $\bigcap_{\alpha < \gamma < \mu} \mathcal{X}_\gamma$ *is saturated relative to* $\mathcal{P} \setminus U_\alpha$*.*

Proof. By Lemma C.20 we obtain that $\bigcap_{\gamma \leq \beta} \mathcal{X}_\gamma$ is saturated relative to

$$\bigcup_{\gamma \leq \beta} A_\gamma = A_0 \cup \bigcup_{\gamma < \beta} A_{\gamma+1} = U_0 \cup \bigcup_{\gamma < \beta} U_{\gamma+1} \setminus U_\gamma = U_\beta \enspace .$$

The same lemma implies that $\bigcap_{\beta < \gamma \leq \alpha} \mathcal{X}_\gamma$ must be saturated relative to

$$\bigcup_{\beta < \gamma \leq \alpha} A_\gamma = \bigcup_{\beta \leq \gamma < \alpha} A_{\gamma+1} = \bigcup_{\beta \leq \gamma < \gamma+1 \leq \alpha} U_{\gamma+1} \setminus U_\gamma = U_\alpha \setminus U_\beta$$

and that $\bigcap_{\alpha < \gamma < \mu} \mathcal{X}_\gamma$ must be saturated relative to

$$\bigcup_{\alpha < \gamma < \mu} A_\gamma = \bigcup_{\alpha \leq \gamma < \gamma+1 < \mu} A_{\gamma+1} = \bigcup_{\alpha \leq \gamma < \gamma+1 < \mu} U_{\gamma+1} \setminus U_\gamma = \mathcal{P} \setminus U_\alpha \enspace .$$ □

Lemma C.23. *Let* $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha < \mu}$ *be a splitting sequence for an MKNF knowledge base* $\mathcal{K}$*,* $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha < \mu}$ *the saturation sequence induced by* $\boldsymbol{U}$*,* $\mathcal{M}$ *an MKNF model of* $\mathcal{K}$ *and* $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha < \mu}$ *a sequence of MKNF interpretations where* $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ *for all* $\alpha < \mu$*. Then for every ordinal* $\alpha < \mu$ *it holds that*

$$\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha) \enspace .$$

Proof. We prove by induction on α:

1° Suppose that $\alpha = 0$. We need to show that $\mathcal{X}_0 = \sigma(\mathcal{M}, U_0)$, which follows directly from the definition of $\mathcal{X}_0$.

2° Take some α such that $\alpha + 1 < \mu$. By the inductive assumption, $\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$. We immediately obtain:

$$\begin{aligned}\bigcap_{\beta \leq \alpha+1} \mathcal{X}_\beta &= \mathcal{X}_{\alpha+1} \cap \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta = \mathcal{X}_{\alpha+1} \cap \sigma(\mathcal{M}, U_\alpha) \\ &= \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha) \cap \sigma(\mathcal{M}, U_\alpha) \enspace .\end{aligned}$$

It remains to show that $\sigma(\mathcal{M}, U_{\alpha+1}) = \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha) \cap \sigma(\mathcal{M}, U_\alpha)$. We know that $U_{\alpha+1}$ is a splitting set for $\mathcal{K}$ and that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, so by Corollary C.16 it follows that $\mathcal{N} = \sigma(\mathcal{M}, U_{\alpha+1})$ is an MKNF model of $b_{U_{\alpha+1}}(\mathcal{K})$. Furthermore, U_α is a splitting set for $b_{U_{\alpha+1}}(\mathcal{K})$, so by another application of Corollary C.16 we obtain that

$$\sigma(\mathcal{M}, U_{\alpha+1}) = \mathcal{N} = \sigma(\mathcal{N}, U_\alpha) \cap \sigma(\mathcal{N}, \mathcal{P} \setminus U_\alpha) \enspace . \tag{C.5}$$

Moreover, Proposition A.23 yields

$$\sigma(\mathcal{N}, U_\alpha) = \sigma(\sigma(\mathcal{M}, U_{\alpha+1}), U_\alpha) = \sigma(\mathcal{M}, U_{\alpha+1} \cap U_\alpha) = \sigma(\mathcal{M}, U_\alpha) \tag{C.6}$$

and

$$\begin{aligned}\sigma(\mathcal{N}, \mathcal{P} \setminus U_\alpha) &= \sigma(\sigma(\mathcal{M}, U_{\alpha+1}), \mathcal{P} \setminus U_\alpha) = \sigma(\mathcal{M}, U_{\alpha+1} \cap (\mathcal{P} \setminus U_\alpha)) \\ &= \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha) \enspace .\end{aligned} \tag{C.7}$$

The desired conclusion follows from (C.5), (C.6) and (C.7).

3° Suppose $\alpha < \mu$ is a limit ordinal and for all $\beta < \alpha$ it holds that $\bigcap_{\gamma \leq \beta} \mathcal{X}_\gamma = \sigma(\mathcal{M}, U_\beta)$. First note that

$$\begin{aligned}\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta &= \mathcal{X}_\alpha \cap \bigcap_{\beta < \alpha} \mathcal{X}_\beta = \mathcal{I} \cap \bigcap_{\beta < \alpha} \bigcap_{\gamma \leq \beta} \mathcal{X}_\gamma = \bigcap_{\beta < \alpha} \sigma(\mathcal{M}, U_\beta) \\ &= \bigcap_{\beta < \alpha} \left\{ I \in \mathcal{I} \;\middle|\; \exists J \in \mathcal{M} : J^{[U_\beta]} = I^{[U_\beta]} \right\} \\ &= \left\{ I \in \mathcal{I} \;\middle|\; \forall \beta < \alpha \; \exists J \in \mathcal{M} : J^{[U_\beta]} = I^{[U_\beta]} \right\}\end{aligned}$$

and also that

$$\sigma(\mathcal{M}, U_\alpha) = \sigma\left(\mathcal{M}, \bigcup_{\beta < \alpha} U_\beta\right) = \left\{ I \in \mathcal{I} \;\middle|\; \exists J \in \mathcal{M} : J^{[\bigcup_{\beta < \alpha} U_\beta]} = I^{[\bigcup_{\beta < \alpha} U_\beta]} \right\} \enspace .$$

From these two identities it can be inferred that $\sigma(\mathcal{M}, U_\alpha)$ is a subset of $\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta$. Indeed, if I belongs to $\sigma(\mathcal{M}, U_\alpha)$, then for some $J \in \mathcal{M}$ we have $J^{[\bigcup_{\beta < \alpha} U_\beta]} = I^{[\bigcup_{\beta < \alpha} U_\beta]}$, hence for any $\beta_0 < \alpha$ and any atom p such that $\mathsf{pr}(p) \subseteq U_{\beta_0} \subseteq \bigcup_{\beta < \alpha} U_\beta$ we obtain

$$J \models p \qquad \text{if and only if} \qquad I \models p \enspace ,$$

which implies that $I^{[U_{\beta_0}]} = J^{[U_{\beta_0}]}$.

To prove the converse inclusion, let $\mathcal{Y} = \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta$ and proceed by contradiction, assuming that $\sigma(\mathcal{M}, U_\alpha) \subsetneq \mathcal{Y}$. By Corollary C.16 we know that $\sigma(\mathcal{M}, U_\alpha)$ is an MKNF model of $b_{U_\alpha}(\mathcal{K})$, so there is some formula $\phi \in b_{U_\alpha}(\mathcal{K})$ such that

$$(\mathcal{Y}, \sigma(\mathcal{M}, U_\alpha)) \not\models \phi \enspace .$$

Furthermore, since $U_\alpha = \bigcup_{\beta < \alpha} U_\beta$ and $\mathsf{pr}(\phi)$ is a finite set of predicate symbols, there is some $\beta < \alpha$ such that $\mathsf{pr}(\phi)$ is a subset of U_β. Consequently, by Corollary A.17, we obtain

$$\left(\mathcal{Y}^{[U_\beta]}, \sigma(\mathcal{M}, U_\alpha)^{[U_\beta]}\right) \not\models \phi \enspace .$$

Let $\mathcal{Y}_1 = \bigcap_{\gamma \leq \beta} \mathcal{X}_\gamma$ and $\mathcal{Y}_2 = \bigcap_{\beta < \gamma \leq \alpha} \mathcal{X}_\gamma$. By Lemma C.22, $\mathcal{Y}_1$ is saturated relative to U_β and $\mathcal{Y}_2$ is saturated relative to $U_\alpha \setminus U_\beta$ and thus by Lemma A.25 also relative to $\mathcal{P} \setminus U_\beta$. Furthermore, $\mathcal{Y} = \mathcal{Y}_1 \cap \mathcal{Y}_2$, so, by Proposition A.28, $\mathcal{Y}^{[U_\beta]} = \mathcal{Y}_1^{[U_\beta]}$. Hence,

$$\left(\mathcal{Y}_1^{[U_\beta]}, \sigma(\mathcal{M}, U_\alpha)^{[U_\beta]}\right) \not\models \phi$$

and the inductive assumption for β yields

$$\left(\sigma(\mathcal{M}, U_\beta)^{[U_\beta]}, \sigma(\mathcal{M}, U_\alpha)^{[U_\beta]}\right) \not\models \phi \enspace .$$

Finally, since U_β is a subset of U_α, Proposition A.24 implies that

$$\sigma(\mathcal{M}, U_\alpha)^{[U_\beta]} = \mathcal{M}^{[U_\beta]} = \sigma(\mathcal{M}, U_\beta)^{[U_\beta]} \enspace .$$

Therefore,

$$\left(\sigma(\mathcal{M}, U_\beta)^{[U_\beta]}, \sigma(\mathcal{M}, U_\beta)^{[U_\beta]}\right) \not\models \phi \enspace .$$

Corollary A.17 now implies that $(\sigma(\mathcal{M}, U_\beta), \sigma(\mathcal{M}, U_\beta)) \not\models \phi$. But at the same time, U_β is a splitting set for $\mathcal{K}$, so, by Corollary C.16, $\sigma(\mathcal{M}, U_\beta)$ is an MKNF model of $b_{U_\beta}(\mathcal{K})$. Since ϕ belongs to $b_{U_\beta}(\mathcal{K})$, we have reached a contradiction. □

Proposition C.24. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$, $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ the saturation sequence induced by $\boldsymbol{U}$, $\mathcal{M}$ an MKNF model of $\mathcal{K}$ and $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ a sequence of MKNF interpretations where $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ for all $\alpha < \mu$. Then $\boldsymbol{X}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$.*

Proof. There are four conditions to verify.

First, $\mathcal{X}_0$ must be an MKNF model of $b_{U_0}(\mathcal{K})$. Since U_0 is a splitting set for $\mathcal{K}$, Corollary C.16 yields that $\sigma(\mathcal{M}, U_0)$ is an MKNF model of $b_{U_0}(\mathcal{K})$. By definition, $\mathcal{X}_0 = \sigma(\mathcal{M}, U_0)$, thus this part of the proof is finished.

Second, for any ordinal α such that $\alpha + 1 < \mu$ it must hold that $\mathcal{X}_{\alpha+1}$ is an MKNF model of

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \enspace .$$

By Corollary C.16, $\mathcal{N} = \sigma(\mathcal{M}, U_{\alpha+1})$ is an MKNF model of $b_{U_{\alpha+1}}(\mathcal{K})$. Furthermore, it can be seen that U_α is a splitting set for $b_{U_{\alpha+1}}(\mathcal{K})$, so by another application of Corollary C.16, we obtain that $\sigma(\mathcal{N}, \mathcal{P} \setminus U_\alpha)$ is an MKNF model of $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K}), \sigma(\mathcal{N}, U_\alpha))$. Moreover, by Proposition A.23,

$$\sigma(\mathcal{N}, U_\alpha) = \sigma(\sigma(\mathcal{M}, U_{\alpha+1}), U_\alpha) = \sigma(\mathcal{M}, U_{\alpha+1} \cap U_\alpha) = \sigma(\mathcal{M}, U_\alpha)$$

and also

$$\begin{aligned}\sigma(\mathcal{N}, \mathcal{P} \setminus U_\alpha) &= \sigma(\sigma(\mathcal{M}, U_{\alpha+1}), \mathcal{P} \setminus U_\alpha) = \sigma(\mathcal{M}, U_{\alpha+1} \cap (\mathcal{P} \setminus U_\alpha)) \\ &= \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha) \enspace .\end{aligned}$$

Lemma C.23 implies that $\sigma(\mathcal{M}, U_\alpha) = \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta$ and since $\mathcal{X}_{\alpha+1} = \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha)$, it follows that $\mathcal{X}_{\alpha+1}$ is an MKNF model of

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \enspace .$$

Third, for every limit ordinal $\alpha < \mu$, $\mathcal{X}_\alpha = \mathcal{I}$ holds by definition.

Fourth, we need to verify that $\bigcap_{\alpha<\mu} \mathcal{X}_\alpha \neq \emptyset$. It follows from the definition of $\boldsymbol{X}$ by Proposition A.22 that $\mathcal{M}$ is a subset of $\mathcal{X}_\alpha$ for every $\alpha < \mu$. Hence,

$$\emptyset \neq \mathcal{M} \subseteq \bigcap_{\alpha<\mu} \mathcal{X}_\alpha \enspace .$$ □

Proposition C.25. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$. If $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$, then for all $\alpha < \mu$, $\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta$ is an MKNF model of $b_{U_\alpha}(\mathcal{K})$.*

Proof. Let $\mathcal{Y}_\alpha = \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta$ for every $\alpha < \mu$. We proceed by induction on α:

1° For $\alpha = 0$ we need to show that $\mathcal{Y}_0 = \mathcal{X}_0$ is an MKNF model of $b_{U_0}(\mathcal{K})$. This follows directly from the assumptions.

2° For α such that $\alpha+1 < \mu$ we need to show that $\mathcal{Y}_{\alpha+1}$ is an MKNF model of $b_{U_{\alpha+1}}(\mathcal{K})$. By the inductive assumption, $\mathcal{Y}_\alpha$ is an MKNF model of $b_{U_\alpha}(\mathcal{K})$. Furthermore,

$$b_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K})) = b_{U_\alpha}(\mathcal{K})$$

and since $\boldsymbol{X}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$, $\mathcal{X}_{\alpha+1}$ is an MKNF model of

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \mathcal{Y}_\alpha\right) \quad .$$

Since U_α is a splitting set for $\mathcal{K}$, it is also a splitting set for $b_{U_{\alpha+1}}(\mathcal{K})$. Consequently, by Theorem C.15, $\mathcal{Y}_\alpha \cap \mathcal{X}_{\alpha+1} = \mathcal{Y}_{\alpha+1}$ is an MKNF model of $b_{U_{\alpha+1}}(\mathcal{K})$.

3° For a limit ordinal $\alpha < \mu$ we need to show that $\mathcal{Y}_\alpha$ is an MKNF model of $b_{U_\alpha}(\mathcal{K})$. First we show that $\mathcal{Y}_\alpha \models b_{U_\alpha}(\mathcal{K})$ and then that for every $\mathcal{Y}' \supsetneq \mathcal{Y}_\alpha$ it holds that $(\mathcal{Y}', \mathcal{Y}_\alpha) \not\models b_{U_\alpha}(\mathcal{K})$.
Take some $\phi \in b_{U_\alpha}(\mathcal{K})$ and suppose that $\beta < \alpha$ is some ordinal such that $\mathsf{pr}(\phi) \subseteq U_\beta$. We know that $\mathcal{Y}_\beta$ is an MKNF model of $b_{U_\beta}(\mathcal{K})$, so $\mathcal{Y}_\beta \models \phi$. Furthermore, for every γ such that $\gamma < \mu$, $\mathcal{X}_{\gamma+1}$ is an MKNF model of $e_{U_\gamma}(b_{U_{\gamma+1}}(\mathcal{K}), \mathcal{Y}_\gamma)$, so by Proposition A.18, $\mathcal{X}_{\gamma+1}$ is saturated relative to $U_{\gamma+1} \setminus U_\gamma$. Consequently, by Lemma C.22, $\mathcal{Y}_\beta = \bigcap_{\gamma \leq \beta} \mathcal{X}_\gamma$ is saturated relative to U_β and $\bigcap_{\beta < \gamma \leq \alpha} \mathcal{X}_\gamma$ is saturated relative to $U_\alpha \setminus U_\beta$ and, by Lemma A.25, it is also saturated relative to $\mathcal{P} \setminus U_\beta$. Hence, by Proposition A.28, for $\mathcal{Y}_\alpha = \mathcal{Y}_\beta \cap \bigcap_{\beta < \gamma \leq \alpha} \mathcal{X}_\gamma$ it holds that $\mathcal{Y}_\alpha^{[U_\beta]} = \mathcal{Y}_\beta^{[U_\beta]}$, and so $\mathcal{Y}_\alpha \models \phi$ follows from Corollary A.14.
Now suppose that $\mathcal{Y}' \supsetneq \mathcal{Y}_\alpha$. Then there must be some $I \in \mathcal{Y}' \setminus \mathcal{Y}_\alpha$. Take some $\beta < \alpha$ such that $I \notin \mathcal{Y}_\beta$ (there must be such β, otherwise $I \in \mathcal{Y}_\alpha$). Let $\mathcal{Y}'' = \mathcal{Y}' \cup \mathcal{Y}_\beta$. By the inductive assumption, $\mathcal{Y}_\beta$ is an MKNF model of $b_{U_\beta}(\mathcal{K})$, so there must be some $\phi \in b_{U_\beta}(\mathcal{K})$ such that $(\mathcal{Y}'', \mathcal{Y}_\beta) \not\models \phi$. Furthermore, $\mathcal{Y}_\beta^{[U_\beta]} = \mathcal{Y}_\alpha^{[U_\beta]}$ and

$$\mathcal{Y}''^{[U_\beta]} = \mathcal{Y}'^{[U_\beta]} \cup \mathcal{Y}_\beta^{[U_\beta]} = \mathcal{Y}'^{[U_\beta]} \cup \mathcal{Y}_\alpha^{[U_\beta]} = (\mathcal{Y}' \cup \mathcal{Y}_\alpha)^{[U_\beta]} = \mathcal{Y}'^{[U_\beta]} \quad .$$

Consequently, by Corollary A.17, $(\mathcal{Y}', \mathcal{Y}_\alpha) \not\models \phi$. □

Lemma C.26. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha < \mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$ and let $\boldsymbol{V} = \langle V_\alpha \rangle_{\alpha < \mu+1}$ be a sequence of sets of predicate symbols such that for every $\alpha < \mu$, $V_\alpha = U_\alpha$ and $V_\mu = \mathcal{P}$. Then $\boldsymbol{V}$ is a splitting sequence for $\mathcal{K}$.*

Moreover, if $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha < \mu}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$, then $\boldsymbol{Y} = \langle \mathcal{Y}_\alpha \rangle_{\alpha < \mu+1}$, where for all $\alpha < \mu$, $\mathcal{Y}_\alpha = \mathcal{X}_\alpha$, and $\mathcal{Y}_\mu = \mathcal{I}$, is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{V}$.

Proof. It is not difficult to verify that $\boldsymbol{V}$ is monotone, continuous, that every V_α is a splitting set for $\mathcal{K}$ and that $\bigcup_{\alpha < \mu+1} V_\alpha = \mathcal{P}$.

Now suppose that $\boldsymbol{X}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$. All the properties of $\boldsymbol{X}$ propagate to $\boldsymbol{Y}$, so one only needs to check that μ is handled correctly. In case μ is a limit ordinal, we need to show that $\mathcal{Y}_\mu = \mathcal{I}$, which holds by definition. On the other hand, if μ is a non-limit ordinal, then let β be such that $\beta + 1 = \mu$. From $\bigcup_{\alpha < \mu} U_\alpha = \mathcal{P}$ it follows that $U_\beta = \mathcal{P}$, so we obtain

$$e_{V_\beta}\left(b_{V_\mu}(\mathcal{K}), \bigcap_{\gamma \leq \beta} \mathcal{Y}_\gamma\right) = e_{\mathcal{P}}\left(\mathcal{K}, \bigcap_{\gamma \leq \beta} \mathcal{Y}_\gamma\right) = \emptyset \quad .$$

Consequently, $\mathcal{Y}_\mu = \mathcal{I}$ is its MKNF model. □

Theorem C.27 (Splitting Sequence Theorem for MKNF Knowledge Bases)**.** *Let $\boldsymbol{U}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$. Then $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ for some solution $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$.*

Proof. Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ and suppose that $\boldsymbol{A}$ is the saturation sequence induced by $\boldsymbol{U}$. If $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, then it follows by Proposition C.24 that there is a solution $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$ where $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ for all $\alpha < \mu$.

Let $\boldsymbol{V} = \langle V_\alpha \rangle_{\alpha<\mu+1}$ and $\boldsymbol{Y} = \langle \mathcal{Y}_\alpha \rangle_{\alpha<\mu+1}$ where $V_\alpha = U_\alpha$ and $\mathcal{Y}_\alpha = \mathcal{X}_\alpha$ for all $\alpha < \mu$, $V_\mu = \mathcal{P}$ and $\mathcal{Y}_\mu = \mathcal{I}$. By Lemma C.26, $\boldsymbol{V}$ is a splitting sequence for $\mathcal{K}$ and $\boldsymbol{Y}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{V}$. Thus, by Lemma C.23,

$$\bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \bigcap_{\alpha<\mu} \mathcal{Y}_\alpha = \mathcal{I} \cap \bigcap_{\alpha<\mu} \mathcal{Y}_\alpha = \bigcap_{\alpha<\mu+1} \mathcal{Y}_\alpha = \sigma(\mathcal{M}, V_\mu) = \sigma(\mathcal{M}, \mathcal{P}) = \mathcal{M} \enspace .$$

To prove the converse implication, suppose that $\boldsymbol{X}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$. Then, by Lemma C.26, there is also a solution $\boldsymbol{Y}$ to $\mathcal{K}$ w.r.t. $\boldsymbol{V} = \langle V_\alpha \rangle_{\alpha<\mu+1}$ such that for all $\alpha < \mu$, $V_\alpha = U_\alpha$ and $V_\mu = \mathcal{P}$. Furthermore,

$$\bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \bigcap_{\alpha\leq\mu} \mathcal{Y}_\alpha$$

and, by Proposition C.25, $\bigcap_{\alpha\leq\mu} \mathcal{Y}_\alpha$ is an MKNF model of $b_{V_\mu}(\mathcal{K}) = b_{\mathcal{P}}(\mathcal{K}) = \mathcal{K}$. □

Corollary C.28. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$, $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ the saturation sequence induced by $\boldsymbol{U}$, $\mathcal{M}$ an MKNF model of $\mathcal{K}$ and $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ a sequence of MKNF interpretations where $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ for all $\alpha < \mu$. Then $\boldsymbol{X}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$ and $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$.*

Proof. Follows from the proof of Theorem C.27 and from Proposition C.24. □

Corollary C.29. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$ such that there exists at least one solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$. Then $\mathcal{K}$ is MKNF satisfiable, and $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ for some solution $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$.*

Proof. Follows by Theorem C.27. □

Corollary C.30. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for an MKNF knowledge base $\mathcal{K}$. An MKNF sentence ϕ is an MKNF consequence of $\mathcal{K}$ if and only if for every solution $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$, $\bigcap_{\alpha<\mu} \mathcal{X}_\alpha \models \phi$.*

Proof. Follows by Theorem C.27. □

Theorem 4.13 (Splitting Theorem for MKNF Knowledge Bases). *The MKNF models semantics for MKNF knowledge bases satisfies the splitting set and splitting sequence properties.*

Proof. Follows from Theorems C.15 and C.27. □

C.1.2 Ontology Updates

Proposition C.31. *Let $\mathcal{T}$ be a first-order theory, $\boldsymbol{U}$ a splitting sequence for $\mathcal{T}$ and $\boldsymbol{A}$ the saturation sequence induced by $\boldsymbol{U}$. Then,*

$$[\![\mathcal{T}]\!] = \bigcap_{\alpha<\mu} [\![b_{A_\alpha}(\mathcal{T})]\!] \enspace .$$

Proof. Since $\boldsymbol{U}$ is a splitting sequence for $\mathcal{T}$, for every formula $\phi \in \mathcal{T}$ there exists a unique $\alpha < \mu$ such that ϕ belongs to $b_{A_\alpha}(\mathcal{T})$. Hence,

$$[\![\mathcal{T}]\!] = \bigcap_{\phi \in \mathcal{T}} [\![\phi]\!] = \bigcap_{\alpha<\mu} \bigcap_{\phi \in b_{A_\alpha}(\mathcal{T})} [\![\phi]\!] = \bigcap_{\alpha<\mu} [\![b_{A_\alpha}(\mathcal{T})]\!] \ . \qquad \square$$

Lemma C.32. *Let $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. Then the following holds:*

(1) *If $\mathcal{M} \subseteq \mathcal{N}$, then $\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \mathcal{M}$.*

(2) *If $\mathcal{M} \supseteq \mathcal{N}$, then $\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \mathcal{N}$.*

(3) *$\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{M} = \mathcal{M}$.*

(4) *$\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \emptyset$ if and only if $\mathcal{M} = \emptyset$ or $\mathcal{N} = \emptyset$.*

Proof.

(1) For every interpretation I, $I <^I_{\mathrm{W}} J$ for every $J \neq I$. Thus, if $\mathcal{M} \subseteq \mathcal{N}$, for all $I \in \mathcal{M}$ it holds that $I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \{ I \}$. Therefore, $\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \mathcal{M}$.

(2) For every interpretation I, $I <^I_{\mathrm{W}} J$ for every $J \neq I$. So if $\mathcal{M} \supseteq \mathcal{N}$, then every interpretation $I \in \mathcal{N} \subseteq \mathcal{M}$ also belongs to $\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}$. Furthermore, $\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} \subseteq \mathcal{N}$ by construction, so we obtain $\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \mathcal{N}$.

(3) Follows from (1).

(4) The direct implication follows by definition of $\blacklozenge_{\mathrm{W}}$. The converse implication follows from (1) and (2). $\square$

Proposition C.33. *Let $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$ be a sequence of first-order theories with $n > 0$ and $\mathcal{M} = [\![\Diamond_{\mathrm{W}} \boldsymbol{T}]\!]$. Then $\mathcal{M} \models \mathcal{T}_{n-1}$.*

Proof. Follows by induction on n, due to the fact that $\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} \subseteq \mathcal{N}$ for any $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. $\square$

Proposition C.34. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence, $I, J \in \mathfrak{I}$ and $\mathcal{N} \in \mathfrak{M}$ be sequence-saturated relative to $\boldsymbol{A}$. Then,*

$$J \in (I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}) \qquad \textit{if and only if} \qquad \forall \alpha < \mu : J^{[A_\alpha]} \in \left(I^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]} \right) \ .$$

Proof. Suppose that $J \notin (I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N})$. If $J \notin \mathcal{N}$, then since $\mathcal{N}$ is sequence-saturated relative to $\boldsymbol{A}$, there is some $\alpha < \mu$ such that $J^{[A_\alpha]} \notin \mathcal{N}^{[A_\alpha]}$. But then $J^{[A_\alpha]} \notin \left(I^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]} \right)$, so we reached the desired conclusion.

In the principal case we have $J \in \mathcal{N}$, so there exists some $K \in \mathcal{N}$ such that $K <^I_{\mathrm{W}} J$. This means that for every predicate symbol $P \in \mathcal{P}$,

$$\left(P^K \div P^I \right) \subseteq \left(P^J \div P^I \right) \tag{C.8}$$

and for some predicate symbol $P_0 \in \mathcal{P}$,

$$\left(P_0^K \div P_0^I \right) \subsetneq \left(P_0^J \div P_0^I \right) \ . \tag{C.9}$$

Since $\boldsymbol{A}$ is a saturation sequence, there is a unique ordinal $\alpha < \mu$ such that $P_0 \in A_\alpha$. It follows from (C.9) that

$$\left(P_0^{K^{[A_\alpha]}} \div P_0^{I^{[A_\alpha]}} \right) = \left(P_0^K \div P_0^I \right) \subsetneq \left(P_0^J \div P_0^I \right) = \left(P_0^{J^{[A_\alpha]}} \div P_0^{I^{[A_\alpha]}} \right) \ .$$

Furthermore, for any predicate symbol $P \in A_\alpha$ it follows from (C.8) that

$$\left(P^{K^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) = \left(P^K \div P^I\right) \subseteq \left(P^J \div P^I\right) = \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) \enspace .$$

Finally, for any predicate symbol P that does not belong to A_α,

$$P^{I^{[A_\alpha]}} = P^{J^{[A_\alpha]}} = P^{K^{[A_\alpha]}} = \emptyset \enspace .$$

Thus,

$$\left(P^{K^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) = \emptyset = \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right)$$

Therefore, we can conclude that

$$K^{[A_\alpha]} <_{\mathsf{W}}^{I^{[A_\alpha]}} J^{[A_\alpha]} \enspace ,$$

so $J^{[A_\alpha]} \notin \left(I^{[A_\alpha]} \mathbin{\blacklozenge_{\mathsf{W}}} \mathcal{N}^{[A_\alpha]}\right)$ as desired.

For the converse implication, suppose that for some $\alpha < \mu$, $J^{[A_\alpha]} \notin \left(I^{[A_\alpha]} \mathbin{\blacklozenge_{\mathsf{W}}} \mathcal{N}^{[A_\alpha]}\right)$. If $J^{[A_\alpha]} \notin \mathcal{N}^{[A_\alpha]}$, we immediately obtain that $J \notin \mathcal{N}$. Consequently, $J \notin (I \mathbin{\blacklozenge_{\mathsf{W}}} \mathcal{N})$.

It remains to consider the principal case when $J^{[A_\alpha]} \in \mathcal{N}^{[A_\alpha]}$. Then there must be some interpretation $K \in \mathcal{N}^{[A_\alpha]}$ such that $K <_{I^{[A_\alpha]}} J^{[A_\alpha]}$. Thus, for all predicate symbols $P \in \mathcal{P}$ we know that

$$\left(P^K \div P^{I^{[A_\alpha]}}\right) \subseteq \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) \enspace . \tag{C.10}$$

We also know that there is some predicate symbol $P_0 \in \mathcal{P}$ such that

$$\left(P_0^K \div P_0^{I^{[A_\alpha]}}\right) \subsetneq \left(P_0^{J^{[A_\alpha]}} \div P_0^{I^{[A_\alpha]}}\right) \enspace . \tag{C.11}$$

Additionally, for every predicate symbol P from $\mathcal{P} \setminus A_\alpha$ it holds that

$$P^{I^{[A_\alpha]}} = P^{J^{[A_\alpha]}} = P^K = \emptyset \enspace .$$

Thus,

$$\left(P^K \div P^{I^{[A_\alpha]}}\right) = \emptyset = \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) \enspace .$$

Consequently, $P_0 \in A_\alpha$. Let K' be an interpretation such that for every ground atom p,

$$K' \models p \qquad \text{if and only if} \qquad K \models p \lor J^{[\mathcal{P} \setminus A_\alpha]} \models p \enspace .$$

It follows that $K'^{[A_\alpha]} = K \in \mathcal{N}^{[A_\alpha]}$ and for every ordinal $\beta < \mu$ such that $\beta \neq \alpha$, $K'^{[A_\beta]} = J^{[A_\beta]} \in \mathcal{N}^{[A_\beta]}$, so since $\mathcal{N}$ is sequence-saturated relative to $\boldsymbol{A}$, $K' \in \mathcal{N}$. Take some predicate symbol $P \in \mathcal{P}$ and consider the following two cases:

a) If $P \in A_\alpha$, then from (C.10) we obtain

$$\left(P^{K'} \div P^I\right) = \left(P^K \div P^{I^{[A_\alpha]}}\right) \subseteq \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) = \left(P^J \div P^I\right) \enspace .$$

b) If $P \in \mathcal{P} \setminus A_\alpha$, then since $K'^{[\mathcal{P} \setminus A_\alpha]} = J^{[\mathcal{P} \setminus A_\alpha]}$,

$$\left(P^{K'} \div P^I\right) = \left(P^J \div P^I\right) \enspace .$$

Moreover, from (C.11) we obtain

$$\left(P_0^{K'} \div P_0^I\right) = \left(P_0^K \div P_0^{I^{[A_\alpha]}}\right) \subsetneq \left(P_0^{J^{[A_\alpha]}} \div P_0^{I^{[A_\alpha]}}\right) = \left(P_0^J \div P_0^I\right) \enspace .$$

It follows from the above considerations that $K' <_{\mathrm{W}}^I J$. Consequently, $J \notin (I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N})$. □

Proposition C.35. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence, $J \in \mathfrak{I}$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ be both sequence-saturated relative to $\boldsymbol{A}$. Then,*

$$J \in (\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}) \qquad \textit{if and only if} \qquad \forall \alpha < \mu : J^{[A_\alpha]} \in \left(\mathcal{M}^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]}\right) \enspace .$$

Proof. By definition, $J \in (\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N})$ if and only if for some $I \in \mathcal{M}$, $J \in (I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N})$. By Proposition C.34, this holds if and only if

$$\forall \alpha < \mu : J^{[A_\alpha]} \in \left(I^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]}\right) \enspace . \tag{C.12}$$

At the same time, the right hand side of our equivalence is true if and only if for some sequence of interpretations $\langle I_\alpha \rangle_{\alpha<\mu}$ the following holds:

$$\forall \alpha < \mu : J^{[A_\alpha]} \in \left(I_\alpha^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]}\right) \enspace . \tag{C.13}$$

It remains to show that (C.12) is equivalent to (C.13). Indeed, (C.12) implies (C.13) by putting $I_\alpha = I$ for all $\alpha < \mu$. Now suppose that (C.13) holds and let I be an interpretation such that for every ground atom p,

$$I \models p \qquad \text{if and only if} \qquad \exists \alpha < \mu : I_\alpha^{[A_\alpha]} \models p \enspace .$$

Then it holds for every $\alpha < \mu$ that $I^{[A_\alpha]} = I_\alpha^{[A_\alpha]} \in \mathcal{M}^{[A_\alpha]}$. Since $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{A}$, this implies that $I \in \mathcal{M}$. Moreover, $J^{[A_\alpha]} \in \left(I^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]}\right)$. As a consequence, (C.12) is satisfied and our proof is finished. □

Proposition C.36. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ be both sequence-saturated relative to $\boldsymbol{A}$. Then,*

$$J^{[A_\alpha]} \in \left(\mathcal{M}^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]}\right) \qquad \textit{if and only if} \qquad J \in (\sigma(\mathcal{M}, A_\alpha) \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)) \enspace .$$

Proof. First note that if $\mathcal{M} = \emptyset$ or $\mathcal{N} = \emptyset$, then the equivalence is trivially satisfied since both $(\mathcal{M}^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]}) = \emptyset$ and $(\sigma(\mathcal{M}, A_\alpha) \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)) = \emptyset$. Thus, we can assume that $\mathcal{M}$ and $\mathcal{N}$ are MKNF interpretations.

By applying Proposition C.35 to $\sigma(\mathcal{M}, A_\alpha)$ and $\sigma(\mathcal{N}, A_\alpha)$ it follows that for every interpretation J,

$$J \in (\sigma(\mathcal{M}, A_\alpha) \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)) \quad \text{if and only if}$$
$$\forall \beta < \mu : J^{[A_\beta]} \in \left(\sigma(\mathcal{M}, A_\alpha)^{[A_\beta]} \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)^{[A_\beta]}\right) \enspace . \tag{C.14}$$

By Lemma A.26 we obtain that whenever $\beta \neq \alpha$, $\sigma(\mathcal{M}, A_\alpha)^{[A_\beta]} = \mathfrak{I}^{[A_\beta]}$ and $\sigma(\mathcal{N}, A_\alpha)^{[A_\beta]} = \mathfrak{I}^{[A_\beta]}$, so by Lemma C.32(3) we can conclude that $\sigma(\mathcal{M}, A_\alpha)^{[A_\beta]} \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)^{[A_\beta]} = \mathfrak{I}^{[A_\beta]}$.

Thus, condition (C.14) gets simplified to

$$J \in (\sigma(\mathcal{M}, A_\alpha) \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)) \quad \text{if and only if} \quad J^{[A_\alpha]} \in \left(\sigma(\mathcal{M}, A_\alpha)^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)^{[A_\alpha]}\right) \enspace .$$

Furthermore, by Proposition A.24, $\sigma(\mathcal{M}, A_\alpha)^{[A_\alpha]} = \mathcal{M}^{[A_\alpha]}$ and $\sigma(\mathcal{N}, A_\alpha)^{[A_\alpha]} = \mathcal{N}^{[A_\alpha]}$, so

$$J \in (\sigma(\mathcal{M}, A_\alpha) \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha)) \qquad \text{if and only if} \qquad J^{[A_\alpha]} \in \left(\mathcal{M}^{[A_\alpha]} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}^{[A_\alpha]}\right) \enspace .$$

This completes our proof. ☐

Corollary C.37. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha < \mu}$ be a saturation sequence and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ be both sequence-saturated relative to $\boldsymbol{A}$. Then,*

$$\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \bigcap_{\alpha < \mu} \sigma(\mathcal{M}, A_\alpha) \mathbin{\blacklozenge_{\mathrm{W}}} \sigma(\mathcal{N}, A_\alpha) \enspace .$$

Proof. Follows from Propositions C.35 and C.36. ☐

Lemma C.38. *Let A be a set of predicate symbols, $I, J \in \mathfrak{I}$ and $\mathcal{N} \in \mathfrak{M}$ be saturated relative to A. If $J \in I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}$, then I coincides with J on $\mathcal{P} \setminus A$.*

Proof. If $J \in I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}$, then $J \in \mathcal{N}$. Let J' be an interpretation such that for every ground atom p,

$$J' \models p \qquad\qquad \text{if and only if} \qquad\qquad J^{[A]} \models p \lor I^{[\mathcal{P} \setminus A]} \models p \enspace .$$

Then $J'^{[A]} = J^{[A]} \in \mathcal{N}$, so since $\mathcal{N}$ is saturated relative to A, J' belongs to $\mathcal{N}$. Furthermore, for any predicate symbol $P \in A$, $P^{J'} \div P^I = P^J \div P^I$ and for any predicate symbol $P \in \mathcal{P} \setminus A$, $P^{J'} \div P^I = \emptyset \subseteq P^J \div P^I$. If this inclusion was proper for some predicate symbol P, then we would obtain that $J' <^I_{\mathrm{W}} J$ holds, contrary to the assumption that J belongs to $I \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}$. Thus, for all predicate symbols $P \in \mathcal{P} \setminus A$, $P^J \div P^I$ must be equal to $\emptyset$. It follows that $J^{[\mathcal{P} \setminus A]} = J'^{[\mathcal{P} \setminus A]} = I^{[\mathcal{P} \setminus A]}$, which is the desired result. ☐

Proposition C.39. *Let $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i < n}$ be a finite sequence of first-order theories, $\boldsymbol{U}$ a splitting sequence for $\boldsymbol{T}$ and $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha < \mu}$ the saturation sequence induced by $\boldsymbol{U}$. Then,*

$$[\![\diamond_{\mathrm{W}} \boldsymbol{T}]\!] = \bigcap_{\alpha < \mu} [\![\diamond_{\mathrm{W}} b_{A_\alpha}(\boldsymbol{T})]\!] \enspace .$$

Proof. We prove by induction on n.

1° If $n = 1$, then $[\![\diamond_{\mathrm{W}} \boldsymbol{T}]\!] = [\![\mathcal{T}_0]\!]$ and for every $\alpha < \mu$, $[\![\diamond_{\mathrm{W}} b_{A_\alpha}(\boldsymbol{T})]\!] = [\![b_{A_\alpha}(\mathcal{T}_0)]\!]$. The claim thus follows from Proposition C.31.

2° We assume that the claim holds for n and prove it for $n + 1$. Let $\boldsymbol{T}' = \langle \mathcal{T}_i \rangle_{i < n+1}$. By definition of $\diamond_{\mathrm{W}}$ we obtain

$$(\diamond_{\mathrm{W}} \boldsymbol{T}') = (\diamond_{\mathrm{W}} \boldsymbol{T}) \diamond_{\mathrm{W}} \mathcal{T}_n \qquad \text{and} \qquad (\diamond_{\mathrm{W}} b_{A_\alpha}(\boldsymbol{T}')) = (\diamond_{\mathrm{W}} b_{A_\alpha}(\boldsymbol{T})) \diamond_{\mathrm{W}} b_{A_\alpha}(\mathcal{T}_n) \enspace .$$

Let $\mathcal{M} = [\![\diamond_{\mathrm{W}} \boldsymbol{T}]\!]$, $\mathcal{N} = [\![\mathcal{T}_n]\!]$ and for every $\alpha < \mu$, $\mathcal{M}_\alpha = [\![\diamond_{\mathrm{W}} b_{A_\alpha}(\boldsymbol{T})]\!]$ and $\mathcal{N}_\alpha = [\![b_{A_\alpha}(\mathcal{T}_n)]\!]$. Our goal is to prove that

$$\mathcal{M} \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N} = \bigcap_{\alpha < \mu} \mathcal{M}_\alpha \mathbin{\blacklozenge_{\mathrm{W}}} \mathcal{N}_\alpha \enspace .$$

We obtain $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha$ and $\mathcal{N} = \bigcap_{\alpha<\mu} \mathcal{N}_\alpha$ by the inductive assumption and by Proposition C.31, respectively. Also, due to Theorem 2.43 and Proposition 2.41, both $\mathcal{M}_\alpha$ and $\mathcal{N}_\alpha$ are saturated relative to A_α.
If $\mathcal{M} = \emptyset$ or $\mathcal{N} = \emptyset$, then, by Lemma C.32(4), $\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N} = \emptyset$ and it follows from Proposition A.35 that for some $\alpha < \mu$, either $\mathcal{M}_\alpha = \emptyset$ or $\mathcal{N}_\alpha = \emptyset$. In either case $\mathcal{M}_\alpha \blacklozenge_{\mathsf{W}} \mathcal{N}_\alpha = \emptyset$ and the desired equation is satisfied.
In the principal case, both $\mathcal{M} \neq \emptyset$ and $\mathcal{N} \neq \emptyset$. Thus, we can use Proposition A.35 to conclude that $\mathcal{M}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ and $\mathcal{N}_\alpha = \sigma(\mathcal{N}, A_\alpha)$ for all $\alpha < \mu$ and, by Proposition A.34, $\mathcal{M}$ and $\mathcal{N}$ are sequence-saturated relative to $\boldsymbol{A}$. Furthermore, we can apply Corollary C.37 to obtain the desired equation:

$$\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N} = \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha) \blacklozenge_{\mathsf{W}} \sigma(\mathcal{N}, A_\alpha) = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha \blacklozenge_{\mathsf{W}} \mathcal{N}_\alpha \enspace . \qquad \square$$

Theorem 4.16 (Splitting Theorem for Winslett's First-Order Operator). *The semantics for sequences of first-order theories induced by Winslett's first-order operator $\diamond_{\mathsf{W}}$ satisfies the splitting set and splitting sequence properties.*

Proof. Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a sequence of first-order theories $\boldsymbol{T}$ and $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be the saturation sequence induced by $\boldsymbol{U}$. The following holds for any $\mathcal{X} \in \mathfrak{M}$ and all ordinals α with $\alpha + 1 < \mu$:

$$\begin{aligned} b_{A_0}(\boldsymbol{T}) &= b_{U_0}(\boldsymbol{T}) \enspace , \\ b_{A_{\alpha+1}}(\boldsymbol{T}) &= b_{U_{\alpha+1} \setminus U_\alpha}(\boldsymbol{T}) = t_{U_\alpha}(b_{U_\alpha+1}(\boldsymbol{T})) = e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{T}), \mathcal{X}) \enspace . \end{aligned}$$

Therefore, the splitting sequence property follows from Proposition C.39. The splitting set property for a splitting set U follows from the splitting sequence property applied to the splitting sequence $\langle U, \mathcal{P} \rangle$. $\square$

C.1.3 Rule Updates

Proposition C.40 (Generalisation of Stable Models). *Let S be one of AS, JU, DS and RD and $\mathcal{P}$ be a logic program. Then J is a stable model of $\mathcal{P}$ if and only if J is an S-model of $\langle \mathcal{P} \rangle$.*

Proof (sketch). In case of the AS- and JU-semantics this follows directly by their definition and by the fact that $\mathsf{rej}_{\mathsf{AS}}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J) = \mathsf{rej}_{\mathsf{JU}}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J) = \emptyset$. In case of the DS-semantics it also holds that $\mathsf{rej}_{\mathsf{DS}}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J) = \emptyset$ and it is not difficult to verify that the equation

$$J' = \mathsf{least}([\mathsf{all}(\langle \mathcal{P} \rangle^{\mathsf{e}}) \setminus \mathsf{rej}_{\mathsf{DS}}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J)] \cup \mathsf{def}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J))$$

is satisfied if and only if J is a stable model of $\mathcal{P}$. Finally, in case of the RD-semantics, if $\mathsf{rej}_{\mathsf{RD}}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J) \neq \emptyset$, then J is not a classical model of $\mathcal{P}$ since it violates at least one of the rules in $\mathcal{P}$. Thus, J is not a stable model of $\mathcal{P}$ and, additionally, the equation

$$J' = \mathsf{least}([\mathsf{all}(\langle \mathcal{P} \rangle^{\mathsf{e}}) \setminus \mathsf{rej}_{\mathsf{RD}}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J)] \cup \mathsf{def}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J)) \enspace ,$$

is not satisfied because for some objective literal l, either $l \in J'$ or $\sim l \in J'$, but neither l nor $\sim l$ appears in heads of rules in the argument program of $\mathsf{least}(\cdot)$, and thus cannot be a part of its least model. On the other hand, if $\mathsf{rej}_{\mathsf{RD}}(\langle \mathcal{P} \rangle^{\mathsf{e}}, J) = \emptyset$, then the same argument as for the DS-semantics applies. $\square$

Proposition C.41. *Let S be one of AS, JU, DS and RD and $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP with $n > 0$. If J is an S-stable model of $\boldsymbol{P}$, then $J \models \mathcal{P}_{n-1}$.*

Proof (sketch). In case of the AS-, JU- and DS-semantics this follows from the fact that rules in $\mathcal{P}_{n-1}$ cannot be rejected. In case of the RD-semantics, if some rules from $\mathcal{P}_{n-1}$ reject each other, then the candidate interpretation is not an RD-model of $\boldsymbol{P}$. □

Theorem 4.20 (Splitting Theorem for Rule Update Semantics). *The rule update semantics AS, JU, DS and RD satisfy the splitting set and splitting sequence properties.*

Proof (sketch). Let S be one of AS, JU, DS and RD. We need to prove that J is an S-model of $\boldsymbol{P}$ if and only if $J = \bigcup_{\alpha<\mu} J_\alpha$ where for every $\alpha < \mu$, J_α is an S-model of $\boldsymbol{P}_\alpha$ and

- $\boldsymbol{P}_0 = b_{U_0}(\boldsymbol{P})$;
- For any ordinal α such that $\alpha + 1 < \mu$, $\boldsymbol{P}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{P}), \bigcup_{\beta\leq\alpha} J_\beta\right)$;
- For any limit ordinal $\alpha < \mu$, $\boldsymbol{P}_\alpha = \langle\emptyset\rangle_{\alpha<\mu}$.

In case S is AS or JU, this follows by the splitting properties of logic programs (Lifschitz and Turner, 1994) and from the observation that rules in $\mathsf{rej}_\mathsf{S}(\boldsymbol{P}, J)$ correspond one-to-one with the rules in $\bigcup_{\alpha<\mu} \mathsf{rej}_\mathsf{S}(\boldsymbol{P}_\alpha, J_\alpha)$. In particular, if we put

$$\mathcal{Q} = [\rho(\boldsymbol{P}) \setminus \mathsf{rej}_\mathsf{S}(\boldsymbol{P}, J)] \qquad \text{and} \qquad \mathcal{Q}_\alpha = [\rho(\boldsymbol{P}_\alpha) \setminus \mathsf{rej}_\mathsf{S}(\boldsymbol{P}_\alpha, J_\alpha)] \enspace ,$$

then it follows that

- $\mathcal{Q}_0 = b_{U_0}(\mathcal{Q})$;
- For every ordinal α such that $\alpha + 1 < \mu$, $\mathcal{Q}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{Q}), \bigcup_{\beta\leq\alpha} J_\beta\right)$;
- For every limit ordinal α, $\mathcal{Q}_\alpha = \emptyset$.

Hence, by the results of (Lifschitz and Turner, 1994), J is a stable model of $\mathcal{Q}$ if and only if $J = \bigcup_{\alpha<\mu} J_\alpha$ where for every $\alpha < \mu$, J_α is a stable model of $\mathcal{Q}_\alpha$. The desired result thus follows by the definition of AS- and JU-models.

A similar argument applies in case S is DS or RD, but the set of default assumptions needs to be handled with additional care. More particularly, if $\boldsymbol{A} = \langle A_\alpha\rangle_{\alpha<\mu}$ is the saturation sequence induced by $\boldsymbol{U}$, then the unconstrained set of default assumptions $\mathsf{def}(\boldsymbol{P}_\alpha, J_\alpha)$ usually introduces rules with predicate symbols outside A_α. In order to overcome this problem, for every DLP $\boldsymbol{Q}$, interpretation K and set of predicate symbols B we introduce the set

$$\mathsf{def}(\boldsymbol{Q}, K, B) = \{\, \sim l \mid l \in \mathcal{L}_\mathsf{G} \wedge \mathsf{pr}(l) \subseteq B \wedge \neg\exists\pi \in \mathsf{all}(\boldsymbol{Q}) : H(\pi) = l \wedge K \models B(\pi) \,\} \enspace .$$

It can be proven that as long as $B \supseteq \mathsf{pr}(\boldsymbol{Q})$, $\mathsf{def}(\boldsymbol{Q}, K)$ can be replaced by $\mathsf{def}(\boldsymbol{Q}, K, B)$ in the definition of an S-model of $\boldsymbol{Q}$, if accompanied by a suitable restriction in the definition of K', i.e. $K' = K \cup \{\, \sim l \mid l \in \mathcal{L}_\mathsf{G} \setminus K \wedge \mathsf{pr}(l) \subseteq B \,\}$. Let

$$\mathcal{Q} = [\rho(\boldsymbol{P}) \setminus \mathsf{rej}_\mathsf{S}(\boldsymbol{P}, J)] \cup \mathsf{def}(\boldsymbol{P}, J, \mathcal{P}) \enspace ,$$
$$\mathcal{Q}_\alpha = [\rho(\boldsymbol{P}_\alpha) \setminus \mathsf{rej}_\mathsf{S}(\boldsymbol{P}_\alpha, J_\alpha)] \cup \mathsf{def}(\boldsymbol{P}_\alpha, J_\alpha, A_\alpha) \enspace .$$

Similarly as before, it follows that

- $\mathcal{Q}_0 = b_{U_0}(\mathcal{Q})$;
- For every ordinal α such that $\alpha + 1 < \mu$, $\mathcal{Q}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{Q}), \bigcup_{\beta\leq\alpha} J_\beta\right)$;
- For every limit ordinal α, $\mathcal{Q}_\alpha = \emptyset$.

Hence, $J' = \mathsf{least}(\mathcal{Q})$ if and only if $J' = \bigcup_{\alpha<\mu} J'_\alpha$ where for every $\alpha < \mu$, $J'_\alpha = \mathsf{least}(\mathcal{Q}_\alpha)$. The desired result thus follows by the definition of DS- and RD-models. □

C.2 Splitting-Based Updates of MKNF Knowledge Bases

Proposition C.42. *Let $\boldsymbol{K}$ be a basic DMKB and $\boldsymbol{U}$ a splitting sequence for $\boldsymbol{K}$ and suppose that both $\diamond$ and S have the splitting sequence property and respect fact update. Then $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ for some solution $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$.*

Proof. Since $\diamond$ and S respect fact update, the case when $\boldsymbol{K}$ is both ontology- and rule-based is treated equivalently by both semantics. Hence, it suffices to consider the case when $\boldsymbol{K}$ is ontology-based and when it is rule-based separately. The same argument applies to the layers of $\boldsymbol{K}$: we do not need to consider cases when, say, $\boldsymbol{K}$ is ontology-based and one of its layers is rule-based, since then the layer is both ontology- and rule-based and the two update semantics coincide when applied to that layer.

Thus, if $\boldsymbol{K}$ is ontology-based, then the claim follows directly from the splitting sequence property of $\diamond$.

On the other hand, if $\boldsymbol{K} = \langle (\emptyset, \mathcal{P}_i) \rangle_{i<n}$ is rule-based, then by the splitting sequence property of S it follows that J is an S-model of $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ if and only if $J = \bigcup_{\alpha<\mu} J_\alpha$ for some solution $\langle J_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{P}$ w.r.t. $\boldsymbol{U}$. Let $\mathcal{M}$ be the MKNF interpretation corresponding to J and for every $\alpha < \mu$, $\mathcal{X}_\alpha$ the MKNF interpretation corresponding to J_α. It follows that

$$\mathcal{M} = \{ I \in \mathfrak{I} \mid I \models J \} \qquad \text{and} \qquad \forall \alpha < \mu : \mathcal{X}_\alpha = \{ I \in \mathfrak{I} \mid I \models J_\alpha \} \enspace .$$

Consequently,

$$\mathcal{M} = \{ I \in \mathfrak{I} \mid I \models J \} = \left\{ I \in \mathfrak{I} \;\middle|\; I \models \textstyle\bigcup_{\alpha<\mu} J_\alpha \right\} = \bigcap_{\alpha<\mu} \{ I \in \mathfrak{I} \mid I \models J_\alpha \} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha \enspace . \qquad \square$$

Proposition C.43. *Let $\boldsymbol{K} = \langle \mathcal{K}_i \rangle_{i<n}$ be a basic DMKB and A a set of predicate symbols such that for all $i < n$, $\mathsf{pr}(\mathcal{K}_i) \subseteq A$. If both $\diamond$ and S conserve the language, then every $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ is saturated relative to A.*

Proof. Let $\mathcal{K}_i = (\mathcal{O}_i, \mathcal{P}_i)$ for all $i < n$. If $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$, then one of the following cases must apply:

a) $\boldsymbol{K}$ is ontology-based and $\mathcal{M} = [\![\diamond \langle \kappa(\mathcal{O}_i) \cup \{ l \mid (l.) \in \mathcal{P}_i \} \rangle_{i<n}]\!]$. The claim then follows from the assumption that $\diamond$ conserves the language.
b) $\boldsymbol{K} = \langle (\emptyset, \mathcal{P}_i) \rangle_{i<n}$ is rule-based and

$$\mathcal{M} = \{ I \in \mathfrak{I} \mid I \models J \}$$

for some S-stable model J of $\langle \mathcal{P}_i \rangle_{i<n}$. We need to prove that $\mathcal{M}$ is saturated relative to A. Take some interpretation I such that $I^{[A]} \in \mathcal{M}^{[A]}$. Then there is some $I' \in \mathcal{M}$ such that $I'^{[A]} = I^{[A]}$. Furthermore, $I' \in \mathcal{M}$ implies that $I' \models J$. Since S conserves the language, we obtain that $\mathsf{pr}(J) \subseteq A$. Consequently, $I \models J$ and, by the definition of $\mathcal{M}$, $I \in \mathcal{M}$. $\square$

Corollary C.44. *Let $\boldsymbol{K}$ be a basic DMKB, $\boldsymbol{U}$ a splitting sequence for $\boldsymbol{K}$ and $\boldsymbol{A}$ the saturation sequence induced by $\boldsymbol{U}$. If both $\diamond$ and S have the splitting sequence property, conserve the language and respect fact update, then every $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ is sequence-saturated relative to $\boldsymbol{A}$.*

Proof. Follows by Propositions C.42, C.43, and A.34. $\square$

Lemma C.45. *Let $\boldsymbol{K}$ be a DMKB and U, V some sets of predicate symbols. Then,*

$$b_U(b_V(\boldsymbol{K})) = b_{U \cap V}(\boldsymbol{K}) \enspace .$$

Proof. Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$. We obtain the following:

$$\begin{aligned} b_U(b_V(\boldsymbol{K})) &= \langle (b_U(b_V(\mathcal{O}_i)), b_U(b_V(\mathcal{P}_i))) \rangle_{i<n} \\ b_{U \cap V}(\boldsymbol{K}) &= \langle (b_{U \cap V}(\mathcal{O}_i), b_{U \cap V}(\mathcal{P}_i)) \rangle_{i<n} \end{aligned}$$

Take some $i < n$. We need to prove that $b_U(b_V(\mathcal{O}_i)) = b_{U \cap V}(\mathcal{O}_i)$ and that $b_U(b_V(\mathcal{P}_i)) = b_{U \cap V}(\mathcal{P}_i)$. By the definition of the bottom of an ontology,

$$\begin{aligned} \phi \in b_U(b_V(\mathcal{O}_i)) &\iff \phi \in b_V(\mathcal{O}_i) \land \mathsf{pr}(\phi) \subseteq U \iff \phi \in \mathcal{O}_i \land \mathsf{pr}(\phi) \subseteq V \land \mathsf{pr}(\phi) \subseteq U \\ &\iff \phi \in \mathcal{O}_i \land \mathsf{pr}(\phi) \subseteq U \cap V \iff \phi \in b_{U \cap V}(\mathcal{O}_i) \enspace . \end{aligned}$$

Similarly, by the definition of the bottom of a program,

$$\begin{aligned} \pi \in b_U(b_V(\mathcal{P}_i)) &\iff \pi \in b_V(\mathcal{P}_i) \land \mathsf{pr}(\pi) \subseteq U \iff \pi \in \mathcal{P}_i \land \mathsf{pr}(\pi) \subseteq V \land \mathsf{pr}(\pi) \subseteq U \\ &\iff \pi \in \mathcal{P}_i \land \mathsf{pr}(\pi) \subseteq U \cap V \iff \pi \in b_{U \cap V}(\mathcal{P}_i) \enspace . \end{aligned}$$ □

Lemma C.46. *Let $\boldsymbol{K}$ be a DMKB, $\mathcal{X} \in \mathcal{M}$, U a set of predicate symbols and V a splitting set for $\boldsymbol{K}$. Then,*

$$e_U(b_V(\boldsymbol{K}), \mathcal{X}) = b_V(e_U(\boldsymbol{K}, \mathcal{X})) \enspace .$$

Proof. Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i)_{i<n} \rangle$. We obtain:

$$\begin{aligned} e_U(b_V(\boldsymbol{K}), \mathcal{X}) &= \langle (t_U(b_V(\mathcal{O}_i)), e_U(b_V(\mathcal{P}_i), \mathcal{X})) \rangle_{i<n} \\ b_V(e_U(\boldsymbol{K}, \mathcal{X})) &= \langle (b_V(t_U(\mathcal{O}_i)), b_V(e_U(\mathcal{P}_i, \mathcal{X}))) \rangle_{i<n} \end{aligned}$$

Take some $i < n$. We need to prove that $t_U(b_V(\mathcal{O}_i)) = b_V(t_U(\mathcal{O}_i))$ and that $e_U(b_V(\mathcal{P}_i), \mathcal{X}) = b_V(e_U(\mathcal{P}_i, \mathcal{X}))$. By the definition of the top and bottom of an ontology,

$$\begin{aligned} \phi \in t_U(b_V(\mathcal{O}_i)) &\iff \phi \in b_V(\mathcal{O}_i) \land \mathsf{pr}(\phi) \nsubseteq U \iff \phi \in \mathcal{O}_i \land \mathsf{pr}(\phi) \subseteq V \land \mathsf{pr}(\phi) \nsubseteq U \\ &\iff \phi \in t_U(\mathcal{O}_i) \land \mathsf{pr}(\phi) \subseteq V \iff \phi \in b_V(t_U(\mathcal{O}_i)) \enspace . \end{aligned}$$

To show that $e_U(b_V(\mathcal{P}_i), \mathcal{X}) \subseteq b_V(e_U(\mathcal{P}_i, \mathcal{X}))$ Take some rule $\sigma \in e_U(b_V(\mathcal{P}_i), \mathcal{X})$. It follows that there is some rule $\pi \in \mathcal{P}_i$ such that $\mathsf{pr}(\pi) \subseteq V$ and

$$\begin{aligned} &H(\sigma) = H(\pi) \enspace , && \mathcal{X} \models \kappa(\{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq U \,\}) \enspace , \\ &B(\sigma) = \{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus U \,\} \enspace , && \mathsf{pr}(\pi) \nsubseteq U \enspace . \end{aligned}$$

Consequently, $\sigma \in e_U(\mathcal{P}_i, \mathcal{X})$ and since $\mathsf{pr}(\pi) \subseteq V$, it follows that $\sigma \in b_V(e_U(\mathcal{P}_i, \mathcal{X}))$.

To show the converse inclusion, take some rule $\sigma \in b_V(e_U(\mathcal{P}_i, \mathcal{X}))$. Then $\mathsf{pr}(\sigma) \subseteq V$ and there exists some rule $\pi \in \mathcal{P}_i$ such that

$$\begin{aligned} &H(\sigma) = H(\pi) \enspace , && \mathcal{X} \models \kappa(\{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq U \,\}) \enspace , \\ &B(\sigma) = \{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus U \,\} \enspace , && \mathsf{pr}(\pi) \nsubseteq U \enspace . \end{aligned}$$

Since $\mathsf{pr}(H(\pi)) \subseteq V$ and V is a splitting set for $\mathcal{P}_i$, it follows that $\mathsf{pr}(\pi) \subseteq V$. Consequently, $\pi \in b_V(\mathcal{P}_i)$ and it follows from the above that $\sigma \in e_U(b_V(\mathcal{P}_i), \mathcal{X})$. □

Lemma C.47. *Let U, V be sets of predicate symbols and $\mathcal{X}, \mathcal{Y}$ some MKNF interpretations such that $\mathcal{X}$ is saturated relative to U, $\mathcal{Y}$ is saturated relative to V and $\mathcal{X}$ coincides with $\mathcal{Y}$ on $U \cap V$. Then,*

$$(\mathcal{X} \cap \mathcal{Y})^{[U]} = \mathcal{X}^{[U]} \qquad \textit{and} \qquad (\mathcal{X} \cap \mathcal{Y})^{[V]} = \mathcal{Y}^{[V]} \enspace .$$

Proof. It suffices to prove one of the equations, the other one follows by the symmetry of the claim. Since $\mathcal{X} \cap \mathcal{Y} \subseteq \mathcal{X}$, it immediately follows that $(\mathcal{X} \cap \mathcal{Y})^{[U]} \subseteq \mathcal{X}^{[U]}$. Take some $I \in \mathcal{X}^{[U]}$. Since $\mathcal{X}$ coincides with $\mathcal{Y}$ on $U \cap V$, there must be some $J \in \mathcal{Y}^{[V]}$ such that $J^{[U \cap V]} = I^{[U \cap V]}$. Let I' be an interpretation such that for every ground atom p,

$$I' \models p \qquad \text{if and only if} \qquad I \models p \vee J \models p \enspace .$$

It follows that $I'^{[U]} = I$ and $I'^{[V]} = J$. From the assumption that $\mathcal{X}$ is saturated relative to U and $\mathcal{Y}$ is saturated relative to V it follows that $I' \in \mathcal{X} \cap \mathcal{Y}$. Hence, we can conclude that $I \in (\mathcal{X} \cap \mathcal{Y})^{[U]}$. ☐

Lemma C.48. *Let U, V be splitting sets for a DMKB $\boldsymbol{K}$ and $\mathcal{X}, \mathcal{Y}$ some MKNF interpretations such that $\mathcal{X}$ is saturated relative to U, $\mathcal{Y}$ is saturated relative to V and $\mathcal{X}$ coincides with $\mathcal{Y}$ on $U \cap V$. Then,*

$$e_U(e_V(\boldsymbol{K}, \mathcal{Y}), \mathcal{X}) = e_{U \cup V}(\boldsymbol{K}, \mathcal{X} \cap \mathcal{Y}) \enspace .$$

Proof. Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$. We obtain:

$$e_U(e_V(\boldsymbol{K}, \mathcal{Y}), \mathcal{X}) = \langle (t_U(t_V(\mathcal{O}_i)), e_U(e_V(\mathcal{P}_i, \mathcal{Y}), \mathcal{X})) \rangle_{i<n}$$
$$e_{U \cup V}(\boldsymbol{K}, \mathcal{X} \cap \mathcal{Y}) = \langle (t_{U \cup V}(\mathcal{O}_i), e_{U \cup V}(\mathcal{P}_i, \mathcal{X} \cap \mathcal{Y})) \rangle_{i<n}$$

Take some $i < n$. We need to prove that $t_U(t_V(\mathcal{O}_i)) = t_{U \cup V}(\mathcal{O}_i)$ and $e_U(e_V(\mathcal{P}_i, \mathcal{Y}), \mathcal{X}) = e_{U \cup V}(\mathcal{P}_i, \mathcal{X} \cap \mathcal{Y})$. Note that whenever U is a splitting set for an ontology $\mathcal{O}$, $t_U(\mathcal{O}) = b_{\mathcal{P} \setminus U}(\mathcal{O})$. Consequently, by Lemma C.45,

$$t_U(t_V(\mathcal{O}_i)) = b_{\mathcal{P} \setminus U}(b_{\mathcal{P} \setminus V}(\mathcal{O}_i)) = b_{(\mathcal{P} \setminus U) \cap (\mathcal{P} \setminus V)}(\mathcal{O}_i) = b_{\mathcal{P} \setminus (U \cup V)}(\mathcal{O}_i) = t_{U \cup V}(\mathcal{O}_i) \enspace .$$

As for the second equation, it holds that $\sigma \in e_U(e_V(\mathcal{P}_i, \mathcal{Y}), \mathcal{X})$ if and only if for some rule $\sigma' \in e_V(\mathcal{P}_i, \mathcal{Y})$,

$$\begin{aligned} &H(\sigma) = H(\sigma') \enspace , && \mathcal{X} \models \kappa(\{\, L \in B(\sigma') \mid \mathsf{pr}(L) \subseteq U \,\}) \enspace , \\ &B(\sigma) = \{\, L \in B(\sigma') \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus U \,\} \enspace , && \mathsf{pr}(\sigma') \nsubseteq U \enspace . \end{aligned} \tag{C.15}$$

Furthermore, $\sigma' \in e_V(\mathcal{P}_i, \mathcal{Y})$ if and only if for some rule $\pi \in \mathcal{P}_i$,

$$\begin{aligned} &H(\sigma') = H(\pi) \enspace , && \mathcal{Y} \models \kappa(\{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq V \,\}) \enspace , \\ &B(\sigma') = \{\, L \in B(\pi) \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus V \,\} \enspace , && \mathsf{pr}(\pi) \nsubseteq V \enspace . \end{aligned} \tag{C.16}$$

Since U and V are splitting sets for $\mathcal{P}_i$, they are also splitting sets for $e_V(\mathcal{P}_i, \mathcal{Y})$ and it follows that $\mathsf{pr}(\sigma') \nsubseteq U$ and $\mathsf{pr}(\pi) \nsubseteq V$ are equivalent to $\mathsf{pr}(H(\sigma')) \nsubseteq U$ and $\mathsf{pr}(H(\pi)) \nsubseteq V$, respectively. Also, since $\mathsf{pr}(H(\pi))$ is a singleton set, together they are equivalent to $\mathsf{pr}(H(\pi)) \nsubseteq U \cup V$.

Moreover, by Lemma C.47, $\mathcal{X} \cap \mathcal{Y}$ coincides with $\mathcal{X}$ on U and with $\mathcal{Y}$ on V. These observations imply that (C.15) and (C.16) are together equivalent to an existence of a rule

$\pi \in \mathcal{P}_i$ such that

$$H(\sigma) = H(\pi) \enspace , \qquad \mathcal{X} \cap \mathcal{Y} \models \kappa(\{ L \in B(\pi) \mid \mathsf{pr}(L) \subseteq U \cup V \}),$$
$$B(\sigma) = \{ L \in B(\pi) \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus (U \cup V) \}, \quad \mathsf{pr}(H(\pi)) \nsubseteq (U \cup V) \enspace .$$

This is equivalent to $\sigma \in e_{U \cup V}(\mathcal{P}_i, \mathcal{X} \cap \mathcal{Y})$. □

Lemma C.49. *Let U, V be sets of predicate symbols, $\mathcal{X} \in \mathcal{M}$ and $\boldsymbol{K}$ a DMKB such that $\mathsf{pr}(\boldsymbol{K}) \subseteq V$. Then,*

$$e_U(\boldsymbol{K}, \mathcal{X}) = e_U(\boldsymbol{K}, \sigma(\mathcal{X}, V))$$

Proof. Note that the second argument of $e_U(\boldsymbol{K}, \cdot)$ is used to interpret literals in bodies of rules in $\boldsymbol{K}$. It follows from Proposition A.24 that $\sigma(\mathcal{X}, V)^{[V]} = \mathcal{X}^{[V]}$. Furthermore, by the assumption it holds for any set of literals S in a body of a rule in $\boldsymbol{K}$ that $\mathsf{pr}(L) \subseteq V$. Thus, by Corollary A.14,

$$\mathcal{X} \models \kappa(S) \iff \sigma(\mathcal{X}, V) \models \kappa(S) \enspace .$$ □

Lemma C.50. *Let $\boldsymbol{A}$ be a saturation sequence, $\mathcal{M} \in \mathcal{M}$ be sequence-saturated relative to $\boldsymbol{A}$ and U a set of predicate symbols. Then $\sigma(\mathcal{M}, U)$ is also sequence-saturated relative to $\boldsymbol{A}$.*

Proof. Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha < \mu}$ and suppose that I is an interpretation such that for every $\alpha < \mu$, $I^{[A_\alpha]} \in \sigma(\mathcal{M}, U)^{[A_\alpha]}$. We need to prove that $I \in \sigma(\mathcal{M}, U)$. It follows from the assumptions that for every $\alpha < \mu$ there exists some $J_\alpha \in \sigma(\mathcal{M}, U)$ such that $I^{[A_\alpha]} = J_\alpha^{[A_\alpha]}$ and some $K_\alpha \in \mathcal{M}$ such that $J_\alpha^{[U]} = K_\alpha^{[U]}$. Let K be an interpretation such that for every ground atom p,

$$K \models p \qquad \text{if and only if} \qquad \exists \alpha < \mu : K_\alpha^{[A_\alpha]} \models p \enspace .$$

It follows that for every $\alpha < \mu$, $K^{[A_\alpha]} = K_\alpha^{[A_\alpha]} \in \mathcal{M}^{[A_\alpha]}$, so since $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{U}$, $K \in \mathcal{M}$. Furthermore, for every $\alpha < \mu$ we obtain that

$$K^{[A_\alpha \cap U]} = K_\alpha^{[A_\alpha \cap U]} = J^{[A_\alpha \cap U]} = I^{[A_\alpha \cap U]} \enspace .$$

Since every predicate symbol from U belongs to A_α for some $\alpha < \mu$, we can conclude that $K^{[U]} = I^{[U]}$. Consequently, $I \in \sigma(\mathcal{M}, U)$. □

Lemma C.51. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha < \mu}$ be a splitting sequence, $\boldsymbol{A}$ the saturation sequence induced by $\boldsymbol{U}$ and $\mathcal{M} \in \mathcal{M}$ be sequence-saturated relative to $\boldsymbol{A}$. Then for any ordinal $\alpha < \mu$,*

$$\sigma\left(\mathcal{M}, \textstyle\bigcup_{\beta \leq \alpha} A_\beta\right) = \sigma(\mathcal{M}, U_\alpha) = \bigcap_{\beta \leq \alpha} \sigma(\mathcal{M}, A_\beta) \enspace .$$

Proof. First we show by induction on α that $U_\alpha = \bigcup_{\beta \leq \alpha} A_\beta$:

1° For $\alpha = 0$ we obtain $U_0 = A_0$ by the definition of $\boldsymbol{A}$.

2° We inductively assume that $U_\alpha = \bigcup_{\beta \leq \alpha} A_\beta$. Then,

$$\bigcup_{\beta \leq \alpha + 1} A_\beta = U_\alpha \cup A_{\alpha+1} = U_\alpha \cup (U_{\alpha+1} \setminus U_\alpha) = U_\alpha \cup U_{\alpha+1} = U_{\alpha+1} \enspace .$$

3° Let α be a limit ordinal. We inductively assume that for all $\beta < \alpha$ it holds that $U_\beta = \bigcup_{\gamma \leq \beta} A_\gamma$. Consequently,

$$\bigcup_{\beta \leq \alpha} A_\beta = A_\alpha \cup \bigcup_{\beta < \alpha} A_\beta = \emptyset \cup \bigcup_{\beta < \alpha} \bigcup_{\gamma \leq \beta} A_\gamma = \bigcup_{\beta < \alpha} U_\beta = U_\alpha \quad .$$

This establishes the first equation.

As for the second equation, if $\mathcal{M}$ is empty, then the claim trivially follows. So suppose that there is some $J_0 \in \mathcal{M}$. If $I \in \sigma(\mathcal{M}, U_\alpha)$, then there is some $J \in \mathcal{M}$ such that $J^{[U_\alpha]} = I^{[U_\alpha]}$. Hence, for every $\beta \leq \alpha$, $J^{[A_\beta]} = I^{[A_\beta]}$. Thus, $I \in \bigcap_{\beta \leq \alpha} \sigma(\mathcal{M}, A_\beta)$. For the other inclusion, let $I \in \bigcap_{\beta \leq \alpha} \sigma(\mathcal{M}, A_\beta)$. Then for every ordinal $\beta \leq \alpha$ there exists some interpretation $I_\beta \in \mathcal{M}$ such that $I_\beta^{[A_\beta]} = I^{[A_\beta]}$. Let J be an interpretation such that for every ground atom p,

$$J \models p \qquad \text{if and only if} \qquad I^{[U_\alpha]} \models p \vee J_0^{[\mathcal{P} \setminus U_\alpha]} \models p \quad .$$

Since $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{A}$, it follows that $J \in \mathcal{M}$. Furthermore, $J^{[U_\alpha]} = I^{[U_\alpha]}$, so $I \in \sigma(\mathcal{M}, U_\alpha)$. □

Proposition 4.27 (Independence of Splitting Sequence). *Let $\boldsymbol{U}$, $\boldsymbol{V}$ be layering splitting sequences for a DMKB $\boldsymbol{K}$. If both $\diamond$ and S have the splitting sequence property, conserve the language and respect fact update, then $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$ if and only if $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ w.r.t. $\boldsymbol{V}$.*

Proof. Suppose $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ w.r.t. $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha < \mu}$. Then $\mathcal{M} = \bigcap_{\alpha < \mu} \mathcal{X}_\alpha$ for some solution $\langle \mathcal{X}_\alpha \rangle_{\alpha < \mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$. This means that:

- $\mathcal{X}_0$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}_0 = b_{U_0}(\boldsymbol{K})$;
- For any ordinal α such that $\alpha + 1 < \mu$, $\mathcal{X}_{\alpha+1}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of

$$\boldsymbol{K}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\gamma \leq \alpha} \mathcal{X}_\gamma\right) \quad ;$$

- For any limit ordinal $\alpha < \mu$, $\mathcal{X}_\alpha = \mathcal{I}$, and thus it is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}_\alpha = \langle \emptyset \rangle$.

Since $\boldsymbol{U}$ is a layering splitting sequence, $\boldsymbol{K}_\alpha$ is a basic DMKB for every $\alpha < \mu$. Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha < \mu}$ be the saturation sequence induced by $\boldsymbol{U}$. We know that for every $\alpha < \mu$, $\boldsymbol{K}_\alpha$ contains only predicate symbols from A_α, so by Proposition C.43, $\mathcal{X}_\alpha$ is saturated relative to A_α. Thus, by Proposition A.35, $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$. Moreover, by Lemma C.51, $\bigcap_{\gamma \leq \alpha} \mathcal{X}_\gamma = \bigcap_{\gamma \leq \alpha} \sigma(\mathcal{M}, A_\gamma) = \sigma(\mathcal{M}, U_\alpha)$, so

$$\boldsymbol{K}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha)\right) \quad .$$

Pick some arbitrary but fixed $\alpha < \mu$ and suppose that $\boldsymbol{V} = \langle V_\beta \rangle_{\beta < \nu}$. Since $\boldsymbol{V}$ is a splitting sequence for $\boldsymbol{K}$, it is also a splitting sequence for $\boldsymbol{K}_\alpha$. Thus, by Proposition C.42 we know that $\mathcal{X}_\alpha = \bigcap_{\beta < \nu} \mathcal{Y}_{\alpha,\beta}$ for some solution $\langle \mathcal{Y}_{\alpha,\beta} \rangle_{\beta < \nu}$ to $\boldsymbol{K}_\alpha$ w.r.t. $\boldsymbol{V}$. Thus,

- $\mathcal{Y}_{\alpha,0}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}_{\alpha,0} = b_{V_0}(\boldsymbol{K}_\alpha)$;
- For any ordinal β such that $\beta + 1 < \nu$, $\mathcal{Y}_{\alpha,\beta+1}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of

$$\boldsymbol{K}_{\alpha,\beta+1} = e_{V_\beta}\left(b_{V_{\beta+1}}(\boldsymbol{K}_\alpha), \textstyle\bigcap_{\gamma \leq \beta} \mathcal{Y}_{\alpha,\gamma}\right) \quad ;$$

- For any limit ordinal $\beta < \nu$, $\mathcal{Y}_{\alpha,\beta} = \mathcal{I}$ and thus it is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}_{\alpha,\beta} = \langle \emptyset \rangle$.

All in all, it follows that for all ordinals α, β such that $\alpha < \mu$ and $\beta < \nu$, $\mathcal{Y}_{\alpha,\beta}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}_{\alpha,\beta}$, as it is defined above.

Since $\boldsymbol{K}_\alpha$ is a basic DMKB, $\boldsymbol{K}_{\alpha,\beta}$ must also be a basic DMKB. Let $\boldsymbol{B} = \langle B_\beta \rangle_{\beta<\nu}$ be the saturation sequence induced by $\boldsymbol{V}$. We know that for every $\beta < \nu$, $\boldsymbol{K}_{\alpha,\beta}$ contains only predicate symbols from B_β, so by Proposition C.43, $\mathcal{Y}_{\alpha,\beta}$ is saturated relative to B_β. Thus, by Propositions A.35 and A.23,

$$\mathcal{Y}_{\alpha,\beta} = \sigma(\mathcal{X}_\alpha, B_\beta) = \sigma(\sigma(\mathcal{M}, A_\alpha), B_\beta) = \sigma(\mathcal{M}, A_\alpha \cap B_\beta) \ .$$

Let the sequence of DMKBs $\boldsymbol{K}' = \langle \boldsymbol{K}'_\beta \rangle_{\beta<\nu}$ be defined as follows:

- $\boldsymbol{K}'_0 = b_{V_0}(\boldsymbol{K})$;
- For any ordinal β such that $\beta + 1 < \nu$,

$$\boldsymbol{K}'_{\beta+1} = e_{V_\beta}\left(b_{V_{\beta+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, V_\beta)\right) \ ;$$

- For any limit ordinal $\beta < \nu$, $\boldsymbol{K}'_\beta = \langle \emptyset \rangle$.

In the following we prove that for any ordinal $\beta < \nu$ and any ordinal α such that $\alpha + 1 < \mu$,

$$\boldsymbol{K}_{0,\beta} = b_{V_0}(\boldsymbol{K}'_\beta) \ , \tag{C.17}$$

$$\boldsymbol{K}_{\alpha+1,\beta} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}'_\beta), \sigma(\mathcal{M}, U_\alpha)\right) \ . \tag{C.18}$$

This is obviously the case whenever β is a limit ordinal, so in the following we consider the cases when it is a non-limit one. Suppose first that $\beta = 0$. Then we can use Lemma C.45 to obtain

$$\boldsymbol{K}_{0,0} = b_{U_0}(\boldsymbol{K}_0) = b_{V_0}(b_{U_0}(\boldsymbol{K})) = b_{U_0 \cap V_0}(\boldsymbol{K}) = b_{U_0}(b_{V_0}(\boldsymbol{K})) = b_{U_0}(\boldsymbol{K}'_0)$$

and for any ordinal α such that $\alpha + 1 < \mu$ we can apply Lemmas C.45 and C.46, achieving the following result:

$$\begin{aligned}
\boldsymbol{K}_{\alpha+1,0} &= b_{V_0}(\boldsymbol{K}_{\alpha+1}) \\
&= b_{V_0}(e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha))) \\
&= e_{U_\alpha}(b_{V_0}(b_{U_{\alpha+1}}(\boldsymbol{K})), \sigma(\mathcal{M}, U_\alpha)) \\
&= e_{U_\alpha}(b_{U_{\alpha+1} \cap V_0}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha)) \\
&= e_{U_\alpha}(b_{U_{\alpha+1}}(b_{V_0}(\boldsymbol{K})), \sigma(\mathcal{M}, U_\alpha)) \\
&= e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}'_0), \sigma(\mathcal{M}, U_\alpha)) \ .
\end{aligned}$$

Now suppose that β is an ordinal such that $\beta + 1 < \nu$. Using Lemmas C.45 and C.46 we obtain:

$$\begin{aligned}\boldsymbol{K}_{0,\beta+1} &= e_{V_\beta}(b_{V_{\beta+1}}(\boldsymbol{K}_0), \sigma(\mathcal{M}, V_\beta)) \\ &= e_{V_\beta}(b_{V_{\beta+1}}(b_{U_0}(\boldsymbol{K})), \sigma(\mathcal{M}, V_\beta)) \\ &= e_{V_\beta}(b_{U_0 \cap V_{\beta+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, V_\beta)) \\ &= e_{V_\beta}(b_{U_0}(b_{V_{\beta+1}}(\boldsymbol{K})), \sigma(\mathcal{M}, V_\beta)) \\ &= b_{U_0}(e_{V_\beta}(b_{V_{\beta+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, V_\beta))) \\ &= b_{U_0}(\boldsymbol{K}'_{\beta+1}) \enspace .\end{aligned}$$

Finally, for any ordinal α such that $\alpha + 1 < \mu$, Lemmas C.45, C.46 and C.48 imply the following:

$$\begin{aligned}\boldsymbol{K}_{\alpha+1,\beta+1} &= e_{V_\beta}(b_{V_{\beta+1}}(\boldsymbol{K}_{\alpha+1}), \sigma(\mathcal{M}, V_\beta)) \\ &= e_{V_\beta}(b_{V_{\beta+1}}(e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha))), \sigma(\mathcal{M}, V_\beta)) \\ &= e_{V_\beta}(e_{U_\alpha}(b_{V_{\beta+1}}(b_{U_{\alpha+1}}(\boldsymbol{K})), \sigma(\mathcal{M}, U_\alpha)), \sigma(\mathcal{M}, V_\beta)) \\ &= e_{U_\alpha \cup V_\beta}(b_{U_{\alpha+1} \cap V_{\beta+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha) \cap \sigma(\mathcal{M}, V_\beta)) \\ &= e_{U_\alpha}(e_{V_\beta}(b_{U_{\alpha+1}}(b_{V_{\beta+1}}(\boldsymbol{K})), \sigma(\mathcal{M}, V_\beta)), \sigma(\mathcal{M}, U_\alpha)) \\ &= e_{U_\alpha}(b_{U_{\alpha+1}}(e_{V_\beta}(b_{V_{\beta+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, V_\beta))), \sigma(\mathcal{M}, U_\alpha)) \\ &= e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}'_{\beta+1}), \sigma(\mathcal{M}, U_\alpha)) \enspace .\end{aligned}$$

Now since $\boldsymbol{K}'_\beta$ is saturated relative to B_β, we can use Lemma C.49 to replace $\sigma(\mathcal{M}, U_\alpha)$ in (C.18) by $\sigma(\mathcal{M}, U_\alpha \cap B_\beta)$. Furthermore, by consecutively using Proposition A.23, Lemmas C.50 and C.51 and Proposition A.23 again, we can see that

$$\begin{aligned}\sigma(\mathcal{M}, U_\alpha \cap B_\beta) = \sigma(\sigma(\mathcal{M}, B_\beta), U_\alpha) &= \bigcap_{\gamma \leq \alpha} \sigma(\sigma(\mathcal{M}, B_\beta), A_\gamma) \\ &= \bigcap_{\gamma \leq \alpha} \sigma(\mathcal{M}, A_\gamma \cap B_\beta) = \bigcap_{\gamma \leq \alpha} \mathcal{Y}_{\gamma,\beta} \enspace .\end{aligned}$$

Hence, (C.18) can be rewritten as:

$$\boldsymbol{K}_{\alpha+1,\beta} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}'_\beta), \textstyle\bigcap_{\gamma \leq \alpha} \mathcal{Y}_{\gamma,\beta}\right) \enspace .$$

Consequently, (C.17) and (C.18) together imply that $\langle \mathcal{Y}_{\alpha,\beta} \rangle_{\alpha<\mu}$ is a solution to $\boldsymbol{K}'_\beta$ for all $\beta < \nu$. Now we can use Proposition C.42 to conclude that

$$\bigcap_{\alpha<\mu} \mathcal{Y}_{\alpha,\beta} = \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha \cap B_\beta) = \bigcap_{\alpha<\mu} \sigma(\sigma(\mathcal{M}, B_\beta), A_\alpha) = \sigma(\mathcal{M}, B_\beta)$$

is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}'_\beta$. One of the last steps in the proof is to show that $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{B}$. We know from Corollary C.44 that $\mathcal{X}_\alpha$ is sequence-saturated relative to $\boldsymbol{B}$, so we obtain the following:

$$\begin{aligned}\bigcap_{\beta<\nu} \sigma(\mathcal{M}, B_\beta) = \bigcap_{\beta<\nu} \bigcap_{\alpha<\mu} \sigma(\sigma(\mathcal{M}, B_\beta), A_\alpha) &= \bigcap_{\alpha<\mu} \bigcap_{\beta<\nu} \sigma(\sigma(\mathcal{M}, A_\alpha), B_\beta) \\ &= \bigcap_{\alpha<\mu} \bigcap_{\beta<\nu} \sigma(\mathcal{X}_\alpha, B_\beta) = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \mathcal{M} \enspace ,\end{aligned}$$

which implies that $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{B}$. Thus, for any $\beta < \nu$, Lemma C.51 implies that

$$\sigma(\mathcal{M}, V_\beta) = \bigcap_{\gamma \leq \beta} \sigma(\mathcal{M}, B_\gamma) \enspace .$$

To sum up, define the sequence of interpretations $\boldsymbol{Z} = \langle \mathcal{Z}_\beta \rangle_{\beta < \nu}$ by $\mathcal{Z}_\beta = \sigma(\mathcal{M}, B_\beta)$. We know the following:

- $\mathcal{Z}_0 = \sigma(\mathcal{M}, B_0)$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}'_0 = b_{V_0}(\boldsymbol{K})$;
- For any ordinal β such that $\beta + 1 < \nu$, $\mathcal{Z}_{\beta+1} = \sigma(\mathcal{M}, B_{\beta+1})$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of

$$\boldsymbol{K}'_{\beta+1} = e_{V_\beta}\left(b_{V_{\beta+1}}(\boldsymbol{K}), \textstyle\bigcap_{\gamma \leq \beta} \sigma(\mathcal{M}, B_\gamma)\right) = e_{V_\beta}\left(b_{V_{\beta+1}}(\boldsymbol{K}), \textstyle\bigcap_{\gamma \leq \beta} \mathcal{Z}_\gamma\right) \enspace ;$$

- For any limit ordinal $\beta < \nu$, put $\mathcal{Z}_\beta = \sigma(\mathcal{M}, B_\beta) = \sigma(\mathcal{M}, \emptyset) = \mathfrak{I}$.

Thus, $\boldsymbol{Z}$ is a solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{V}$. Moreover, since $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{B}$, it follows by Proposition A.34 that

$$\mathcal{M} = \bigcap_{\beta < \nu} \sigma(\mathcal{M}, B_\beta) = \bigcap_{\beta < \nu} \mathcal{Z}_\beta \enspace .$$

So $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ w.r.t. $\boldsymbol{V}$.

Proof of the converse implication follows by the symmetry of the claim. □

C.3 Properties

Theorem 4.30 (Faithfulness w.r.t. MKNF Knowledge Bases). *Suppose that S is faithful to the stable models semantics and let $\langle \mathcal{K} \rangle$ be a layered DMKB. An MKNF interpretation $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ if and only if $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\langle \mathcal{K} \rangle$.*

Proof. Follows by Theorem 4.13 and Propositions 2.17 and 2.19. □

Theorem 4.31 (Faithfulness w.r.t. First-Order Update Operator). *Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \emptyset) \rangle_{i<n}$ be a DMKB. An MKNF interpretation $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M} = [\![\diamond \langle \mathcal{O}_i \rangle_{i<n}]\!]$.*

Proof. Follows by the fact that $\boldsymbol{K}$ is basic, so $\boldsymbol{U} = \langle \mathcal{P} \rangle$ is a layering splitting sequence for $\boldsymbol{K}$. Thus, by Proposition 4.27, $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M} = [\![\diamond \langle \mathcal{O}_i \rangle_{i<n}]\!]$. □

Theorem 4.32 (Faithfulness w.r.t. Rule Update Semantics). *Let $\boldsymbol{K} = \langle (\emptyset, \mathcal{P}_i) \rangle_{i<n}$ be a DMKB. If J is an S-model of $\langle \mathcal{P}_i \rangle_{i<n}$, then the MKNF interpretation corresponding to J is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$. If $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$, then the ASP interpretation corresponding to $\mathcal{M}$ is an S-model of $\langle \mathcal{P}_i \rangle_{i<n}$.*

Proof. Follows by the fact that $\boldsymbol{K}$ is basic, so $\boldsymbol{U} = \langle \mathcal{P} \rangle$ is a layering splitting sequence for $\boldsymbol{K}$. Thus, by Proposition 4.27, $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M}$ corresponds some S-stable model of $\langle \mathcal{P}_i \rangle_{i<n}$. □

Theorem 4.33 (Primacy of New Information). *Suppose that $\diamond$ satisfies (FO1) and S respects primacy of new information and let $\boldsymbol{K} = \langle \mathcal{K}_i \rangle_{i<n}$ be a layered DMKB such that $n > 0$. If $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$, then $\mathcal{M} \models \kappa(\mathcal{K}_{n-1})$.*

Proof. If $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$, then for some layering splitting sequence $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ for $\boldsymbol{K}$, $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ for some solution $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$. This means that

- $\mathcal{X}_0$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $b_{U_0}(\boldsymbol{K})$;
- For any ordinal α such that $\alpha + 1 < \mu$, $\mathcal{X}_{\alpha+1}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \quad ;$$

- For any limit ordinal α, $\mathcal{X}_\alpha = \mathcal{I}$.

Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be the saturation sequence induced by $\boldsymbol{U}$. It follows from the assumptions that $\diamond$ and S conserve the language, $\diamond$ satisfies (FO1) and S respects primacy of new information by Proposition C.43 that

- $\mathcal{X}_0$ is saturated relative to A_0 and $\mathcal{X}_0 \models b_{U_0}(\mathcal{K}_{n-1})$;
- For any ordinal α such that $\alpha + 1 < \mu$, $\mathcal{X}_{\alpha+1}$ is saturated relative to $A_{\alpha+1}$ and

$$\mathcal{X}_{\alpha+1} \models e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}_{n-1}), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \quad ;$$

- For any limit ordinal α, $\mathcal{X}_\alpha = \mathcal{I}$ is saturated relative to $A_\alpha = \emptyset$.

Thus, by Proposition A.34, $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{A}$, by Proposition A.35, $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$, and by Lemma C.51, $\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$.

Now let ϕ be some formula from $\kappa(\mathcal{K}_{n-1})$. If ϕ is of the form $\mathbf{K}\,\psi$ where ψ is a first-order formula, then there must exist a unique ordinal α such that $\mathsf{pr}(\phi) \subseteq A_\alpha$. Due to the above considerations we can then conclude that $\mathcal{X}_\alpha \models \phi$. Furthermore,

$$\mathcal{X}_\alpha \models \phi \iff \sigma(\mathcal{M}, A_\alpha) \models \phi \iff \sigma(\mathcal{M}, A_\alpha)^{[A_\alpha]} \models \phi \iff \mathcal{M}^{[A_\alpha]} \models \phi \iff \mathcal{M} \models \phi \enspace .$$

On the other hand, if $\phi = \kappa(\pi)$ for some rule π, then the there exists a unique non-limit ordinal α such that $\mathsf{pr}(H(\pi)) \subseteq A_\alpha$. Suppose that $\mathcal{M} \models \kappa(B(\pi))$. If $\alpha = 0$, then it follows that $\pi \in b_{U_0}(\mathcal{K}_{n-1})$ and it follows that $\mathcal{X}_0 \models \kappa(B(\pi))$. Consequently, $\mathcal{X}_0 \models \kappa(H(\pi))$ and we can conclude that $\mathcal{M} \models \kappa(H(\pi))$. If $\alpha = \beta + 1$, then there is a rule $\sigma \in e_{U_\beta}(b_{U_{\beta+1}}(\mathcal{K}_{n-1}), \bigcap_{\gamma\leq\beta} \mathcal{X}_\gamma)$ and $\mathcal{X}_{\beta+1} \models \kappa(B(\sigma))$, Consequently, $\mathcal{X}_{\beta+1} \models \kappa(H(\sigma))$ and we obtain $\mathcal{M} \models \kappa(H(\sigma))$. In either case, $\mathcal{M} \models \phi$. □

Definition C.52 (Generalised Update Semantics for Basic DMKBs with Static Rules)**.** Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$ be a basic DMKB with static rules. An MKNF interpretation $\mathcal{M}$ is a *generalised* $(\diamond, \mathsf{S})$*-dynamic MKNF model* of $\boldsymbol{K}$ if either

a) $\boldsymbol{K}$ is ontology-based and $\mathcal{M} = [\![\diamond \langle \kappa(\mathcal{O}_i) \cup \{ l \mid (l.) \in \mathcal{P}_i \} \rangle_{i<n}]\!]$, or
b) $\boldsymbol{K}$ is rule-based and $\mathcal{M}$ corresponds to some ASP interpretation J such that $\mathsf{pr}(J) \subseteq \mathsf{pr}(\mathcal{P}_0)$ and $J \models \mathcal{P}_0$.

Definition C.53 (Generalised Solution to a DMKB with Static Rules)**.** Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$ be a DMKB with static rules and $\boldsymbol{U}$ a layering splitting sequence for $\boldsymbol{K}$. A *generalised solution to* $\boldsymbol{K}$ *w.r.t.* $\boldsymbol{U}$ is a sequence of MKNF interpretations $\langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ such that

1. $\mathcal{X}_0$ is a generalised $(\diamond, \mathsf{S})$-dynamic MKNF model of $b_{U_0}(\boldsymbol{K})$;

2. For any ordinal α such that $\alpha + 1 < \mu$, $\mathcal{X}_{\alpha+1}$ is a generalised $(\diamond, \mathsf{S})$-dynamic MKNF model of
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right) \enspace ;$$
3. For any limit ordinal α, $\mathcal{X}_\alpha = \mathfrak{I}$;
4. $\bigcap_{\alpha<\mu} \mathcal{X}_\alpha \neq \emptyset$.

Remark C.54. For the remainder of this section we assume that $\diamond$ has the splitting sequence property, conserves the language, respects fact update and satisfies (FO2.⊤) as well as (FO8.2), and S is faithful to the stable models semantics.

Proposition C.55. *Let $\boldsymbol{K}$ be a positive DMKB with static rules and $\boldsymbol{U}$ a layering splitting sequence for $\boldsymbol{K}$. If $\boldsymbol{X}$ is a solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$, then it is a generalised solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$.*

Proof. This follows from the assumption that S is faithful to the stable models semantics and for every stable model J or a program $\mathcal{P}$, $J \models \mathcal{P}$. □

Proposition C.56. *Let $\boldsymbol{K}$ be a positive DMKB with static rules and $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ a layering splitting sequence for $\boldsymbol{K}$. If there is a generalised solution $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$, then there is a solution $\boldsymbol{Y} = \langle \mathcal{Y}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$ such that for all $\alpha < \mu$, $\mathcal{X}_\alpha \subseteq \mathcal{Y}_\alpha$.*

Proof. Let $\boldsymbol{Y} = \langle \mathcal{Y}_\alpha \rangle_{\alpha<\mu}$ be as follows:

- If $b_{U_0}(\boldsymbol{K})$ is ontology-based, then
$$\mathcal{Y}_0 = \left[\!\left[\diamond \left\langle \kappa(b_{U_0}(\mathcal{O}_i)) \cup \{\, l \mid (l.) \in b_{U_0}(\mathcal{P}_i) \,\} \right\rangle_{i<n} \right]\!\right] \enspace .$$
Otherwise, $\mathcal{Y}_0$ corresponds to the least set of objective literals J such that
$$J \models b_{U_0}(\mathcal{P}_0) \enspace .$$
- For any ordinal α such that $\alpha + 1 < \mu$, if
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \leq \alpha} \mathcal{Y}_\beta\right)$$
is ontology-based, then
$$\mathcal{Y}_{\alpha+1} = \left[\!\left[\diamond \left\langle \kappa(t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i))) \cup \left\{\, l \;\middle|\; (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \bigcap_{\beta \leq \alpha} \mathcal{Y}_\beta\right) \,\right\} \right\rangle_{i<n} \right]\!\right] \enspace .$$
Otherwise, $\mathcal{Y}_{\alpha+1}$ corresponds to the least set of objective literals J such that
$$J \models e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \leq \alpha} \mathcal{Y}_\beta\right) \enspace .$$
- For any limit ordinal α, $\mathcal{Y}_\alpha = \mathfrak{I}$.

We verify by induction on α that $\mathcal{X}_\alpha \subseteq \mathcal{Y}_\alpha$:

1° For $\alpha = 0$ we consider two cases. If $b_{U_0}(\boldsymbol{K})$ is ontology-based, then clearly $\mathcal{X}_0 = \mathcal{Y}_0$. If it is rule-based, then $\mathcal{X}_0$ corresponds to some ASP interpretation J such that $J \models b_{U_0}(\mathcal{P}_0)$, and $\mathcal{Y}_0$ to the least set of literals J' such that $J' \models b_{U_0}(\mathcal{P}_0)$. Clearly, $J \supseteq J'$, so
$$\mathcal{X}_0 = \{\, I \in \mathfrak{I} \mid I \models J \,\} \subseteq \{\, I \in \mathfrak{I} \;\middle|\; I \models J' \,\} = \mathcal{Y}_0 \enspace .$$

2° Assuming that the claim holds for all $\beta \leq \alpha$, we prove that $\mathcal{X}_{\alpha+1} \subseteq \mathcal{Y}_{\alpha+1}$. Let $\mathcal{X} = \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta$ and $\mathcal{Y} = \bigcap_{\beta \leq \alpha} \mathcal{Y}_\beta$. By the inductive assumption, $\mathcal{X} \subseteq \mathcal{Y}$. Consequently,

$$e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{X}) \supseteq e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{Y}) \ . \tag{C.19}$$

If $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{X})$ is ontology-based, then $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{X})$ and $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{Y})$ differ only in the first component. According to (C.19), the first component of $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{X})$ is a superset of the first component of $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{Y})$. Thus, due to (FO8.2), we can conclude that $\mathcal{X}_{\alpha+1} \subseteq \mathcal{Y}_{\alpha+1}$.
If $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{X})$ is rule-based, then $\mathcal{X}_{\alpha+1}$ corresponds to an ASP interpretation J such that $J \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{X})$ and $\mathcal{Y}_{\alpha+1}$ corresponds to a the minimal set of objective literals J' such that $J' \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{Y})$. It then follows from (C.19) that $J \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{Y})$, so $J \supseteq J'$ and we can conclude that

$$\mathcal{X}_{\alpha+1} = \{\, I \in \mathcal{I} \mid I \models J \,\} \subseteq \{\, I \in \mathcal{I} \mid I \models J' \,\} = \mathcal{Y}_{\alpha+1} \ .$$

3° The case when α is a limit ordinal follows trivially from $\mathcal{X}_\alpha = \mathcal{Y}_\alpha = \mathcal{I}$.

We have shown that $\mathcal{Y}_\alpha \neq \emptyset$ for every $\alpha < \mu$ and it follows by the definition of $\boldsymbol{Y}$ and the assumption that S is faithful to the stable models semantics that $\boldsymbol{Y}$ is a solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$. □

Corollary C.57. *Let $\boldsymbol{K}$ be a positive DMKB with static rules and $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ a layering splitting sequence for $\boldsymbol{K}$. Then either there is no generalised solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$, or there is a unique solution $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$ and for every generalised solution $\boldsymbol{Y} = \langle \mathcal{Y}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$, $\mathcal{Y}_\alpha \subseteq \mathcal{X}_\alpha$ for every $\alpha < \mu$.*

Proof. Follows from Propositions C.55 and C.56. □

Proposition C.58. *Let $\boldsymbol{K}$ be a positive DMKB with static rules, $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ a layering splitting sequence for $\boldsymbol{K}$, $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ a generalised solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$ and $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$. Then,*

$$\mathcal{M} \subseteq T^\diamond_{\boldsymbol{K}}(\mathcal{M}) \ .$$

Proof. Let $\boldsymbol{T} = \langle T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0), \kappa(\mathcal{O}_1), \dots, \kappa(\mathcal{O}_{n-1}) \rangle$ where

$$T_{\mathcal{P}_0}(\mathcal{M}) = \bigcup \{\, H(\pi) \mid \pi \in \mathcal{P}_0 \wedge \mathcal{M} \models \kappa(B(\pi)) \,\} \ .$$

By the definition of $T^\diamond_{\boldsymbol{K}}$,

$$T^\diamond_{\boldsymbol{K}}(\mathcal{M}) = [\![\diamond \boldsymbol{T}]\!] \ .$$

Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be the saturation sequence induced by $\boldsymbol{U}$. By the splitting sequence property of $\diamond$ it follows that

$$T^\diamond_{\boldsymbol{K}}(\mathcal{M}) = [\![\diamond \boldsymbol{T}]\!] = \bigcap_{\alpha<\mu} [\![\diamond b_{A_\alpha}(\boldsymbol{T})]\!] \ .$$

Furthermore, it follows from the definition of generalised solution that $\mathcal{X}_\alpha$ is saturated relative to A_α, so by Proposition A.35 and Lemma C.51,

$$\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha) \qquad \text{and} \qquad \bigcap_{\beta\leq\alpha} \mathcal{X}_\alpha = \sigma(\mathcal{M}, U_\alpha) \ .$$

We prove that for all $\alpha < \mu$, $\mathcal{X}_\alpha \subseteq [\![\diamond b_{A_\alpha}(\boldsymbol{T})]\!]$:

1. For $\alpha = 0$ we obtain $U_0 = A_0$. Suppose first that $b_{U_0}(\boldsymbol{K})$ is ontology-based and $\mathcal{X}_0 = [\![\Diamond \boldsymbol{T}']\!]$ where
$$\boldsymbol{T}' = \langle \kappa(b_{U_0}(\mathcal{O}_i)) \cup \{ l \mid (l.) \in b_{U_0}(\mathcal{P}_i) \} \rangle_{i<n} \enspace .$$
By the assumption that U_0 is a splitting set for $\mathcal{P}_0$ it follows that $b_{A_0}(T_{\mathcal{P}_0}(\mathcal{M})) = \{ l \mid (l.) \in b_{U_0}(\mathcal{P}_0) \}$ and since $\mathcal{P}_i = \emptyset$ for all $i > 0$, this implies that $b_{A_0}(\boldsymbol{T}) = \boldsymbol{T}'$ and we can conclude that $\mathcal{X}_0 = [\![\Diamond b_{A_0}(\boldsymbol{T})]\!]$.
On the other hand, if $b_{U_0}(\boldsymbol{K})$ is rule-based and $\mathcal{X}_0$ corresponds to some ASP interpretation J such that $J \models b_{U_0}(\mathcal{P}_0)$, then by (FO2.⊤),
$$[\![\Diamond b_{A_0}(\boldsymbol{T})]\!] = [\![\Diamond \langle b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \dots, \emptyset \rangle]\!] = [\![b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \enspace .$$
We need to prove that $\mathcal{X}_0 \models b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M}))$. Take some literal $l \in b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M}))$. Then there is some rule $\pi \in \mathcal{P}_0$ such that $H(\pi) = l$, $\mathcal{M} \models \kappa(B(\pi))$ and $\mathsf{pr}(l) \subseteq U_0$. Since U_0 is a splitting set for $\mathcal{P}_0$, $\mathsf{pr}(B(\pi)) \subseteq U_0$. Consequently, $\mathcal{X}_0 = \sigma(\mathcal{M}, U_0) \models \kappa(B(\pi))$ and we obtain that $J \models B(\pi)$. This implies that $J \models l$ and we can conclude that $\mathcal{X}_0 \models l$. Thus, $\mathcal{X}_0 \models b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M}))$.
2. For a non-limit ordinal $\alpha + 1$ we have $A_{\alpha+1} = U_{\alpha+1} \setminus U_\alpha$. Suppose first that
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right)$$
is ontology-based and $\mathcal{X}_{\alpha+1} = [\![\Diamond \boldsymbol{T}']\!]$ where
$$\boldsymbol{T}' = \left\langle \kappa(t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i))) \cup \left\{ l \;\middle|\; (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \textstyle\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right) \right\} \right\rangle_{i<n} \enspace .$$
First note that since U_α and $U_{\alpha+1}$ are splitting sets for $\mathcal{O}_i$, we obtain
$$\begin{aligned} t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i)) &= b_{\mathcal{P} \setminus U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i)) = b_{(\mathcal{P} \setminus U_\alpha) \cap U_{\alpha+1}}(\mathcal{O}_i) = b_{U_{\alpha+1} \setminus U_\alpha}(\mathcal{O}_i) \\ &= b_{A_{\alpha+1}}(\mathcal{O}_i) \enspace . \end{aligned}$$
Furthermore, since $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta)$ contains only facts, every rule $\pi \in \mathcal{P}_0$ such that $\mathsf{pr}(H(\pi)) \subseteq U_{\alpha+1}$ satisfies $\mathsf{pr}(B(\pi)) \subseteq U_\alpha$ and due to the fact that $\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$ we can conclude that
$$b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})) = \left\{ l \;\middle|\; (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \textstyle\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right) \right\} \enspace .$$
Since $\mathcal{P}_i = \emptyset$ for all $i > 0$, the above considerations imply that $b_{A_{\alpha+1}}(\boldsymbol{T}) = \boldsymbol{T}'$ and we can conclude that $\mathcal{X}_{\alpha+1} = [\![\Diamond b_{A_{\alpha+1}}(\boldsymbol{T})]\!]$.
On the other hand, if
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right)$$
is rule-based and $\mathcal{X}_{\alpha+1}$ corresponds to some ASP interpretation J such that $J \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta)$, then by (FO2.⊤),
$$[\![\Diamond b_{A_{\alpha+1}}(\boldsymbol{T})]\!] = [\![\Diamond \left\langle b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \dots, \emptyset \right\rangle]\!] = [\![b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \enspace .$$
We need to prove that $\mathcal{X}_{\alpha+1} \models b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M}))$. Take some literal $l \in b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M}))$. Then there is some rule $\pi \in \mathcal{P}_0$ such that $H(\pi) = l$, $\mathcal{M} \models \kappa(B(\pi))$ and $\mathsf{pr}(l) \subseteq A_{\alpha+1}$. It follows that there is some rule $\sigma \in e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta)$ with $H(\sigma) = l$,

$\mathcal{M} \models \kappa(B(\sigma))$ and $\mathsf{pr}(\sigma) \subseteq A_{\alpha+1}$. Consequently, $\mathcal{X}_{\alpha+1} = \sigma(\mathcal{M}, A_{\alpha+1}) \models \kappa(B(\sigma))$ and so $J \models B(\sigma)$. This implies that $J \models l$ and we can conclude that $\mathcal{X}_{\alpha+1} \models l$. Thus, $\mathcal{X}_{\alpha+1} \models b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M}))$.

3. If α is a limit ordinal, then it follows from (FO2.⊤) that

$$\mathcal{X}_\alpha = \mathcal{I} = [\![\emptyset]\!] = [\![\diamond\langle\emptyset, \emptyset, \ldots, \emptyset\rangle]\!] = [\![\diamond\, b_\emptyset(\boldsymbol{T})]\!] = [\![\diamond\, b_{A_\alpha}(\boldsymbol{T})]\!] \ . \qquad \square$$

Proposition C.59. *Let $\boldsymbol{K}$ be a positive DMKB with static rules, $\boldsymbol{U} = \langle U_\alpha\rangle_{\alpha<\mu}$ a layering splitting sequence for $\boldsymbol{K}$, $\boldsymbol{A} = \langle A_\alpha\rangle_{\alpha<\mu}$ the saturation sequence induced by $\boldsymbol{U}$ and $\mathcal{M}$ an MKNF interpretation. If $\mathcal{M} = T^{\diamond}_{\boldsymbol{K}}(\mathcal{M})$, then $\mathcal{M} = \bigcap_{\alpha<\mu}\sigma(\mathcal{M}, A_\alpha)$ and $\langle\sigma(\mathcal{M}, A_\alpha)\rangle_{\alpha<\mu}$ is a generalised solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$.*

Proof. Let $\boldsymbol{T} = \langle T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0), \kappa(\mathcal{O}_1), \ldots, \kappa(\mathcal{O}_{n-1})\rangle$ where

$$T_{\mathcal{P}_0}(\mathcal{M}) = \bigcup \{\, H(\pi) \mid \pi \in \mathcal{P}_0 \wedge \mathcal{M} \models \kappa(B(\pi)) \,\} \ .$$

By the definition of $T^{\diamond}_{\boldsymbol{K}}$, $T^{\diamond}_{\boldsymbol{K}}(\mathcal{M}) = [\![\diamond\,\boldsymbol{T}]\!]$. By the splitting sequence property of $\diamond$ it follows that

$$\mathcal{M} = T^{\diamond}_{\boldsymbol{K}}(\mathcal{M}) = [\![\diamond\,\boldsymbol{T}]\!] = \bigcap_{\alpha<\mu} [\![\diamond\, b_{A_\alpha}(\boldsymbol{T})]\!] \ .$$

Let $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ for all $\alpha < \mu$. Since $\diamond$ conserves the language, it follows from Proposition A.35 that for all $\alpha < \mu$,

$$\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha) = [\![\diamond\, b_{A_\alpha}(\boldsymbol{T})]\!] \ .$$

We need to prove that

1. $\mathcal{X}_0$ is a generalised $(\diamond, \mathsf{S})$-dynamic MKNF model of $b_{U_0}(\boldsymbol{K})$;
2. For any ordinal α such that $\alpha + 1 < \mu$, $\mathcal{X}_{\alpha+1}$ is a generalised $(\diamond, \mathsf{S})$-dynamic MKNF model of
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\beta\leq\alpha}\mathcal{X}_\beta\right) \ ;$$
3. For any limit ordinal α, $\mathcal{X}_\alpha = \mathcal{I}$;
4. $\bigcap_{\alpha<\mu}\mathcal{X}_\alpha \neq \emptyset$.

We prove each condition separately:

1. Note first that $U_0 = A_0$. If $b_{U_0}(\boldsymbol{K})$ is ontology-based, then we need to prove that $\mathcal{X}_0 = [\![\diamond\,\boldsymbol{T}']\!]$ where

$$\boldsymbol{T}' = \langle \kappa(b_{U_0}(\mathcal{O}_i)) \cup \{\, l \mid (l.) \in b_{U_0}(\mathcal{P}_i) \,\}\rangle_{i<n} \ .$$

Since $\mathcal{P}_i = \emptyset$ for all $i > 0$, it follows that $b_{U_0}(\boldsymbol{T}) = \boldsymbol{T}'$ and thus

$$\mathcal{X}_0 = [\![\diamond\, b_{A_0}(\boldsymbol{T})]\!] = [\![\diamond\, b_{U_0}(\boldsymbol{T})]\!] = [\![\diamond\,\boldsymbol{T}']\!] \ .$$

On the other hand, if $b_{U_0}(\boldsymbol{K})$ is rule-based, then we have to show that $\mathcal{X}_0$ corresponds to some ASP interpretation J such that $J \models b_{U_0}(\mathcal{P}_0)$. Note that due to (FO2.⊤),

$$\mathcal{X}_0 = [\![\diamond\, b_{A_0}(\boldsymbol{T})]\!] = [\![\diamond\langle b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \ldots, \emptyset\rangle]\!] = [\![b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \ .$$

Put $J = \{\, l \in \mathcal{L}_{\mathsf{G}} \mid \mathcal{X}_0 \models l \,\}$. It follows that if $J \models B(\pi)$ for some rule $\pi \in b_{U_0}(\mathcal{P}_0)$, then $\mathcal{X}_0 \models \kappa(B(\pi))$ and thus $\mathcal{M} \models \kappa(B(\pi))$. Consequently, $H(\pi) \in T_{\mathcal{P}_0}(\mathcal{M})$ and we conclude that $\mathcal{X}_0 \models H(\pi)$, thus also $J \models H(\pi)$.

2. For a non-limit ordinal $\alpha + 1$ we have $A_{\alpha+1} = U_{\alpha+1} \setminus U_\alpha$. Suppose first that

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right)$$

is ontology-based. We have to prove that $\mathcal{X}_{\alpha+1} = [\![\diamond \boldsymbol{T}']\!]$ where

$$\boldsymbol{T}' = \left\langle \kappa(t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i))) \cup \left\{ l \;\middle|\; (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \right\}\right\rangle_{i<n} \ .$$

First note that since U_α and $U_{\alpha+1}$ are splitting sets for $\mathcal{O}_i$, we obtain

$$\begin{aligned} t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i)) &= b_{\mathcal{P}\setminus U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i)) = b_{(\mathcal{P}\setminus U_\alpha)\cap U_{\alpha+1}}(\mathcal{O}_i) = b_{U_{\alpha+1}\setminus U_\alpha}(\mathcal{O}_i) \\ &= b_{A_{\alpha+1}}(\mathcal{O}_i) \ . \end{aligned}$$

Furthermore, since $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta)$ contains only facts, every rule $\pi \in \mathcal{P}_0$ such that $\mathsf{pr}(H(\pi)) \subseteq U_{\alpha+1}$ satisfies $\mathsf{pr}(B(\pi)) \subseteq U_\alpha$ and due to the fact that $\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$ we can conclude that

$$b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})) = \left\{ l \;\middle|\; (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \right\} \ .$$

Since $\mathcal{P}_i = \emptyset$ for all $i > 0$, the above considerations imply that $b_{A_{\alpha+1}}(\boldsymbol{T}) = \boldsymbol{T}'$ and we can conclude that

$$\mathcal{X}_{\alpha+1} = [\![\diamond b_{A_0}(\boldsymbol{T})]\!] = [\![\diamond \boldsymbol{T}']\!] \ .$$

On the other hand, if

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right)$$

is rule-based, we need to prove that $\mathcal{X}_{\alpha+1}$ corresponds to some ASP interpretation J such that $J \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta)$. Due to (FO2.⊤),

$$\mathcal{X}_{\alpha+1} = [\![\diamond b_{A_{\alpha+1}}(\boldsymbol{T})]\!] = [\![\diamond \langle b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \dots, \emptyset\rangle]\!] = [\![b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \ .$$

Put $J = \{ l \in \mathcal{L}_\mathsf{G} \mid \mathcal{X}_{\alpha+1} \models l \}$. It follows that if $J \models B(\pi)$ for some rule $\pi \in e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta)$, then there is a rule $\sigma \in b_{U_{\alpha+1}}(\mathcal{P}_0)$ such that $H(\pi) = H(\sigma)$ and $\mathcal{M} \models \kappa(B(\sigma))$. Consequently, $H(\pi) \in T_{\mathcal{P}_0}(\mathcal{M})$ and we obtain $\mathcal{X}_{\alpha+1} \models H(\pi)$, thus also $J \models H(\pi)$.

3. Note that if α is a limit ordinal, then $A_\alpha = \emptyset$. Since $\mathcal{M}$ is an MKNF interpretation, it follows that $\mathcal{M} \neq \emptyset$, and we obtain

$$\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha) = \sigma(\mathcal{M}, \emptyset) = \mathfrak{I} \ .$$

4. Since $\mathcal{M}$ is an MKNF interpretation, it follows that $\emptyset \neq \mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$. □

Proposition C.60. *Let $\boldsymbol{K}$ be a positive layered DMKB with static rules. An MKNF interpretation $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M}$ is a $(\diamond, \boldsymbol{S})$-dynamic MKNF model of $\boldsymbol{K}$.*

Proof. Suppose that $\mathcal{M}$ is the $\diamond$-dynamic MKNF model of $\boldsymbol{K}$. Then $\mathcal{M}$ is the greatest fixed point of $T^{\diamond}_{\boldsymbol{K}}$ and it follows from Proposition C.59 that $\langle \sigma(\mathcal{M}, A_\alpha)\rangle_{\alpha<\mu}$ is a generalised solution to $\boldsymbol{K}$ w.r.t. some splitting sequence $\boldsymbol{U}$. Thus, by Corollary C.57, there exists a solution $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$ such that for all $\alpha < \mu$, $\sigma(\mathcal{M}, A_\alpha) \subseteq \mathcal{X}_\alpha$. Let $\mathcal{N} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$. It holds that $\mathcal{N}$ is the $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ and we obtain

$$\mathcal{M} = \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha) \subseteq \bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \mathcal{N} \ .$$

Also, it follows from Propositions C.55 and C.58 that

$$\mathcal{N} \subseteq T^{\diamond}_{\boldsymbol{K}}(\mathcal{N})$$

and since $\mathcal{M}$ is the greatest fixed point of $T^{\diamond}_{\boldsymbol{K}}$, we can conclude that $\mathcal{N} \subseteq \mathcal{M}$.

Similarly, if $\mathcal{N}$ is the $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$, then there is a solution $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. some layering splitting sequence $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha<\mu}$ such that

$$\mathcal{N} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha \ .$$

By Propositions C.55 and C.58, $\mathcal{N} \subseteq T^{\diamond}_{\boldsymbol{K}}(\mathcal{N})$ and we can conclude that $\mathcal{N} \subseteq \mathcal{M}$ where $\mathcal{M}$ is the greatest fixed point of $T^{\diamond}_{\boldsymbol{K}}$, i.e. the $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$. It follows from Proposition C.59 that $\langle \sigma(\mathcal{M}, A_\alpha) \rangle_{\alpha<\mu}$ is a generalised solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$. Thus, by Corollary C.57, there exists a solution $\boldsymbol{X} = \langle \mathcal{X}_\alpha \rangle_{\alpha<\mu}$ to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$ such that for all $\alpha < \mu$, $\sigma(\mathcal{M}, A_\alpha) \subseteq \mathcal{X}_\alpha$. It follows from the uniqueness of a $\diamond$-dynamic MKNF model that $\mathcal{N} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha$ and we obtain

$$\mathcal{M} = \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha) \subseteq \bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \mathcal{N} \ .$$ □

Proposition C.61. *Let $\boldsymbol{K}$ be a layered DMKB with static rules. An MKNF interpretation $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}^{\mathcal{M}}$.*

Proof. Follows from the assumption that S is faithful to the stable models semantics and by the definition of a stable model. □

Theorem 4.34 (Compatibility with Update Semantics from Chapter 3). *Suppose that $\diamond$ satisfies (FO2.⊤) and (FO8.2) and that S is faithful to the stable models semantics. Let $\boldsymbol{K}$ be a layered DMKB with static rules. An MKNF interpretation $\mathcal{M}$ is a $\diamond$-dynamic MKNF model of $\boldsymbol{K}$ if and only if $\mathcal{M}$ is a $(\diamond, \mathsf{S})$-dynamic MKNF model of $\boldsymbol{K}$.*

Proof. Follows by the definition of a $\diamond$-dynamic MKNF model and by Propositions C.60 and C.61. □

D

Proofs: Belief Updates on SE-Models

In the following we present proofs of results from Chapter 6, implicitly working under the same assumptions as those imposed in that chapter. That is, we constrain ourselves to propositional logic programs without explicit negation over a finite set of propositional atoms $\mathcal{A}$.

D.1 Semantic Rule Update Operators Based on SE-Models

Definition D.1 (Program Corresponding to a Set of Three-Valued Interpretations)**.** Let $\mathcal{M}$ be a set of three-valued interpretations. We denote by $\|\mathcal{M}\|$ some arbitrary but fixed program $\mathcal{P}$ such that

$$[\![\mathcal{P}]\!]_{\mathsf{SE}} = \{\, X, X^* \mid X \in \mathcal{M} \,\} \enspace .$$

Instead of $\|\,\{\, X_1, X_2, \ldots, X_n \,\}\,\|$ we usually write $\|X_1, X_2, \ldots, X_n\|$.

Definition D.2 (Order Assignment Generated by an Update Operator)**.** Let $\oplus$ be a rule update operator and X an three-valued interpretation. We define the binary relation $\prec_{\oplus}^{X}$ for all three-valued interpretations Y, Z as follows: $Y \prec_{\oplus}^{X} Z$ if and only if the following conditions are satisfied:

$$Y \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}} \tag{D.1}$$

$$Z \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}} \tag{D.2}$$

$$\text{If } Y \neq Y^*, \text{ then } Z \in [\![\, \|X\| \oplus \|Y^*, Z\| \,]\!]_{\mathsf{SE}} \tag{D.3}$$

The *preorder assignment generated by* $\oplus$ assigns to every three-valued interpretation X the reflexive and transitive closure $\leq_{\oplus}^{X}$ of $\prec_{\oplus}^{X}$, i.e. $Y \leq_{\oplus}^{X} Z$ if and only if $Y = Z$ or there is some $n \geq 2$ and three-valued interpretations $Y_1, Y_2, \ldots, Y_n$ such that $Y = Y_1 \prec_{\oplus}^{X} Y_2 \prec_{\oplus}^{X} \cdots \prec_{\oplus}^{X} Y_n = Z$.

Lemma D.3. *Let $\oplus$ be a rule update operator satisfying conditions (P1)$_{\mathsf{SE}}$ – (P8)$_{\mathsf{SE}}$ and X, Y, Z some three-valued interpretations. If $Y \leq_{\oplus}^{X} Z$, then either $Y = Z$ or $Z \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.*

Proof. Suppose that $Y \neq Z$. Then, by the definition of $\leq_{\oplus}^{X}$, for some $n \leq 2$ and three-valued interpretations $Y_1, Y_2, \ldots, Y_n$ it holds that $Y = Y_1 \prec_{\oplus}^{X} Y_2 \prec_{\oplus}^{X} \cdots \prec_{\oplus}^{X} Y_n = Z$. We will prove by induction on n that $Y_n \notin [\![\|X\| \oplus \|Y_1, Y_n\|]\!]_{\mathsf{SE}}$ from which the desired result follows directly.

1° For $n = 2$ this follows from $Y_1 \prec_{\oplus}^{X} Y_2$ by (D.2).

2° We inductively assume that

$$Y_n \notin [\![\|X\| \oplus \|Y_1, Y_n\|]\!]_{\mathsf{SE}} \tag{D.4}$$

and prove that $Y_{n+1} \notin [\![\|X\| \oplus \|Y_1, Y_{n+1}\|]\!]_{\mathsf{SE}}$.
We know that $Y_n \prec_{\oplus}^{X} Y_{n+1}$, so by (D.2) we obtain

$$Y_{n+1} \notin [\![\|X\| \oplus \|Y_n, Y_{n+1}\|]\!]_{\mathsf{SE}} \enspace . \tag{D.5}$$

Considering that the program $\|Y_1, Y_n, Y_{n+1}\| \mathbin{\dot{\wedge}} \|Y_1, Y_n\|$ is strongly equivalent to $\|Y_1, Y_n\|$, by (P5)$_{\mathsf{SE}}$ and (P4)$_{\mathsf{SE}}$ we conclude that

$$(\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|) \mathbin{\dot{\wedge}} \|Y_1, Y_n\| \models_{\mathsf{SE}} \|X\| \oplus \|Y_1, Y_n\|$$

which, together with (D.4), implies that

$$Y_n \notin [\![\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|]\!]_{\mathsf{SE}} \enspace . \tag{D.6}$$

Similarly, since the program $\|Y_1, Y_n, Y_{n+1}\| \mathbin{\dot{\wedge}} \|Y_n, Y_{n+1}\|$ is strongly equivalent to $\|Y_n, Y_{n+1}\|$, by (P5)$_{\mathsf{SE}}$ and (P4)$_{\mathsf{SE}}$ we obtain that

$$(\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|) \mathbin{\dot{\wedge}} \|Y_n, Y_{n+1}\| \models_{\mathsf{SE}} \|X\| \oplus \|Y_n, Y_{n+1}\| \enspace ,$$

and so due to (D.5) it holds that

$$Y_{n+1} \notin [\![\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|]\!]_{\mathsf{SE}} \enspace . \tag{D.7}$$

Now we consider two cases:

a) If $Y_n = Y_n^*$, then (D.6) and (P1)$_{\mathsf{SE}}$ imply that

$$\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_{n+1}\| \enspace ;$$
$$\|X\| \oplus \|Y_1, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_n, Y_{n+1}\| \enspace ,$$

so by (P6)$_{\mathsf{SE}}$ we can conclude that $\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|$ is strongly equivalent to $\|X\| \oplus \|Y_1, Y_{n+1}\|$. But then the desired conclusion follows from (D.7).

b) If $Y_n \neq Y_n^*$, then from (D.3) we infer that

$$Y_{n+1} \in [\![\|X\| \oplus \|Y_n^*, Y_{n+1}\|]\!]_{\mathsf{SE}} \enspace . \tag{D.8}$$

Furthermore, from (D.6) and (P1)$_{\mathsf{SE}}$ we obtain

$$\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_n^*, Y_{n+1}\| \enspace ;$$
$$\|X\| \oplus \|Y_1, Y_n^*, Y_{n+1}\| \models_{\mathsf{SE}} \|Y_1, Y_n, Y_{n+1}\| \enspace ,$$

so by (P6)$_{\mathsf{SE}}$ we can conclude that $\|X\| \oplus \|Y_1, Y_n, Y_{n+1}\|$ is strongly equivalent to $\|X\| \oplus \|Y_1, Y_n^*, Y_{n+1}\|$ and, due to (D.7),

$$Y_{n+1} \notin [\![\, \|X\| \oplus \|Y_1, Y_n^*, Y_{n+1}\| \,]\!]_{\mathsf{SE}} \enspace .$$

Since $\|Y_1, Y_n^*, Y_{n+1}\|$ is strongly equivalent to $\|Y_1, Y_{n+1}\| \dot{\vee} \|Y_n^*, Y_{n+1}\|$, it follows from (P4)$_{\mathsf{SE}}$ and (P7)$_{\mathsf{SE}}$ that either $Y_{n+1} \notin [\![\, \|X\| \oplus \|Y_1, Y_{n+1}\| \,]\!]_{\mathsf{SE}}$ or $Y_{n+1} \notin [\![\, \|X\| \oplus \|Y_n^*, Y_{n+1}\| \,]\!]_{\mathsf{SE}}$. The latter is impossible due to (D.8). □

Lemma D.4. *Let $\oplus$ be a rule update operator satisfying conditions (P1)$_{\mathsf{SE}}$ – (P8)$_{\mathsf{SE}}$ and X, Y, Z, some three-valued interpretations. If $Y \not\prec_\oplus^X Z$, then the following conditions are satisfied:*

(1) If $Y = Z^*$, then $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

(2) If $Y = Y^*$ and $Z \in [\![\, \|X\| \oplus \|Z\| \,]\!]_{\mathsf{SE}}$, then $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

(3) If $Y \neq Y^*$ and $Z \in [\![\, \|X\| \oplus \|Y^*, Z\| \,]\!]_{\mathsf{SE}}$, then $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

Proof. First we show the following auxiliary statement: If $Y = Z$ or $Y \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$, then all three conditions are satisfied.

First suppose that $Y = Z$. If $Y = Z^*$, then $Y = Y^* = Z = Z^*$, so it follows from (P1)$_{\mathsf{SE}}$ and (P3)$_{\mathsf{SE}}$ that $[\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}} = [\![\, \|X\| \oplus \|Z^*\| \,]\!]_{\mathsf{SE}} = \{ Z^* \}$, verifying condition (1). Furthermore, conditions (2) and (3) are satisfied because $\|Z\| = \|Y^*, Z\| = \|Y, Z\|$.

Now suppose that $Y \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$. If $Y = Z^*$, then it follows from (P1)$_{\mathsf{SE}}$ and (P3)$_{\mathsf{SE}}$ that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$. If $Y = Y^*$, then it follows from (P1)$_{\mathsf{SE}}$ that

$$\|X\| \oplus \|Y, Z\| \models_{\mathsf{SE}} \|Z\| \qquad \text{and} \qquad \|X\| \oplus \|Z\| \models_{\mathsf{SE}} \|Y, Z\| \enspace ,$$

so by (P6)$_{\mathsf{SE}}$ we obtain that $\|X\| \oplus \|Y, Z\| \equiv_{\mathsf{SE}} \|X\| \oplus \|Z\|$. Hence it follows from $Z \in [\![\, \|X\| \oplus \|Z\| \,]\!]_{\mathsf{SE}}$ that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$. On the other hand, if $Y \neq Y^*$, then it follows from (P1)$_{\mathsf{SE}}$ that

$$\|X\| \oplus \|Y, Z\| \models_{\mathsf{SE}} \|Y^*, Z\| \qquad \text{and} \qquad \|X\| \oplus \|Y^*, Z\| \models_{\mathsf{SE}} \|Y, Z\| \enspace ,$$

so by (P6)$_{\mathsf{SE}}$ we obtain that $\|X\| \oplus \|Y, Z\| \equiv_{\mathsf{SE}} \|X\| \oplus \|Y^*, Z\|$. Hence it follows from $Z \in [\![\, \|X\| \oplus \|Y^*, Z\| \,]\!]_{\mathsf{SE}}$ that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

Turning to the proof of the lemma, note that since $Y \not<_\oplus^X Z$, either $Y \not\preceq_\oplus^X Z$ or $Z \leq_\oplus^X Y$. In the former case, $Y \not\prec_\oplus^X Z$, so, by the definition of $\prec_\oplus^X$, either $Y \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$, so we can apply our auxiliary statement, or $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$ as desired, or $Y \neq Y^*$ and $Z \notin [\![\, \|X\| \oplus \|Y^*, Z\| \,]\!]_{\mathsf{SE}}$, in which case all three conditions are trivially satisfied. In the latter case it follows from Lemma D.3 that either $Y = Z$ or $Y \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$, so the rest follows once again from the auxiliary statement. □

Proposition D.5. *Let $\oplus$ be a rule update operator satisfying conditions (P1)$_{\mathsf{SE}}$ – (P8)$_{\mathsf{SE}}$, X an three-valued interpretation and $\mathcal{U}$ a program. Then,*

$$[\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}} = \min \left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_\oplus^X \right) \enspace .$$

Proof. If $[\![\mathcal{U}]\!]_{\mathsf{SE}} = \emptyset$, then $[\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}}$ is empty by (P1)$_{\mathsf{SE}}$ and the equation holds trivially.

If $[\![\mathcal{U}]\!]_{\mathsf{SE}} \neq \emptyset$, then by (P3)$_{\mathsf{SE}}$ there is some $Z \in [\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}}$ and by (P1)$_{\mathsf{SE}}$, $Z \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$. Suppose that Z is not minimal in $[\![\mathcal{U}]\!]_{\mathsf{SE}}$ w.r.t. $\leq_\oplus^X$. Then there is some $Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$ such that $Y <_\oplus^X Z$. Thus, $Y \neq Z$, so by Lemma D.3 we conclude that $Z \notin [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$. Considering that $\mathcal{U} \dot{\wedge} \|Y, Z\|$ is strongly equivalent to $\|Y, Z\|$, it follows from (P4)$_{\mathsf{SE}}$ and (P5)$_{\mathsf{SE}}$ that $(\|X\| \oplus \mathcal{U}) \dot{\wedge} \|Y, Z\| \models_{\mathsf{SE}} \|X\| \oplus \|Y, Z\|$. Consequently, $Z \notin [\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}}$, contrary to our assumption. Therefore, $[\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}}$ is a subset of $\min([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_\oplus^X)$.

To prove the converse inclusion, assume that Z is minimal in $[\![\mathcal{U}]\!]_{\mathsf{SE}}$ w.r.t. $\leq_{\oplus}^{X}$ and take some $Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$. Note that $Y \not<_{\oplus}^{X} Z$, so we can use Lemma D.4. We will show that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$. We consider three cases:

a) If $Y = Z^*$, then $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$ follows immediately from condition (1) of Lemma D.4.

b) If $Y = Y^*$, then we can conclude from the previous case and from the fact that $[\![\mathcal{U}]\!]_{\mathsf{SE}}$ is well-defined that $Z \in [\![\, \|X\| \oplus \|Z\| \,]\!]_{\mathsf{SE}}$. Thus, by condition (2) of Lemma D.4 it follows that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

c) If $Y \neq Y^*$, then we can conclude from the previous case and from the fact that $[\![\mathcal{U}]\!]_{\mathsf{SE}}$ is well-defined that $Z \in [\![\, \|X\| \oplus \|Y^*, Z\| \,]\!]_{\mathsf{SE}}$. Thus, by condition (3) of Lemma D.4 it follows that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$.

The choice of Y was arbitrary, so we have proven that $Z \in [\![\, \|X\| \oplus \|Y, Z\| \,]\!]_{\mathsf{SE}}$ for all $Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$. This means that by repeated application of (P7)$_{\mathsf{SE}}$, Z is an SE-model of the program

$$\|X\| \oplus \dot{\bigvee_{Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}}} \|Y, Z\|$$

and since $\mathcal{U}$ is strongly equivalent to the program $\dot{\bigvee}_{Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}} \|Y, Z\|$, it follows from (P4)$_{\mathsf{SE}}$ that $Z \in [\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}}$. $\square$

Proposition D.6. *If a rule update operator $\oplus$ satisfies conditions (P1)$_{\mathsf{SE}}$ – (P8)$_{\mathsf{SE}}$, then the preorder assignment generated by $\oplus$ is semi-faithful and organised and it characterises $\oplus$.*

Proof. We first show that the assignment generated by $\oplus$ characterises $\oplus$. We know that $\mathcal{P}$ is strongly equivalent to the program $\dot{\bigvee}_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \|X\|$, so by (P4)$_{\mathsf{SE}}$ and repeated application of (P8)$_{\mathsf{SE}}$ we obtain that $\mathcal{P} \oplus \mathcal{U}$ is strongly equivalent to the program

$$\dot{\bigvee_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}}} (\|X\| \oplus \mathcal{U}) \enspace .$$

Furthermore, by Proposition D.5 it follows that $[\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}} = \min \left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\oplus}^{X} \right)$, so indeed

$$[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} [\![\, \|X\| \oplus \mathcal{U} \,]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min \left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\oplus}^{X} \right) \enspace . \tag{D.9}$$

To see that the assignment generated by $\oplus$ is semi-faithful, first take some three-valued interpretations X, Y such that $Y \neq X$ and $Y \neq X^*$. We need to show that either $X <_{\oplus}^{X} Y$ or $X^* <_{\oplus}^{X} Y$. The equation (D.9) together with (P2)$_{\mathsf{SE}}$ imply that

$$\begin{aligned}
[\![\, \|X\| \oplus \|Y^*, X\| \,]\!]_{\mathsf{SE}} &= \min \left(\{ Y^*, X, X^* \}, \leq_{\oplus}^{X} \right) \cup \min \left(\{ Y^*, X, X^* \}, \leq_{\oplus}^{X^*} \right) \\
&= \{ X, X^* \} \enspace , \\
[\![\, \|X\| \oplus \|Y, X\| \,]\!]_{\mathsf{SE}} &= \min \left(\{ Y, Y^*, X, X^* \}, \leq_{\oplus}^{X} \right) \cup \min \left(\{ Y, Y^*, X, X^* \}, \leq_{\oplus}^{X^*} \right) \\
&= \{ X, X^* \} \enspace .
\end{aligned}$$

Thus, Y^* is not minimal within $\{ Y^*, X, X^* \}$ and Y is not minimal within $\{ Y, Y^*, X, X^* \}$

w.r.t. $\leq_{\oplus}^{X}$. In other words:

$$\text{either } X <_{\oplus}^{X} Y^* \text{ or } X^* <_{\oplus}^{X} Y^* \text{ and} \tag{D.10}$$
$$\text{either } X <_{\oplus}^{X} Y \text{ or } X^* <_{\oplus}^{X} Y \text{ or } Y^* <_{\oplus}^{X} Y \enspace . \tag{D.11}$$

In case of the first two alternatives of (D.11), we have already achieved our goal. The third alternative together with (D.10) and transitivity of $<_{\oplus}^{X}$ also concludes the proof of the first condition of semi-faithfulness. To see that the second condition holds as well, consider that by (P2)$_{\text{SE}}$, $[\![\|X^*\| \oplus \|X\|]\!]_{\text{SE}} = \{ X^* \}$ and $[\![\|X\| \oplus \|X\|]\!]_{\text{SE}} = \{ X, X^* \}$, so it follows from (D.9) that

$$X \notin \min(\{ X, X^* \}, \leq_{\oplus}^{X^*}) \quad \text{and} \quad X \in \min(\{ X, X^* \}, \leq_{\oplus}^{X}) \cup \min(\{ X, X^* \}, \leq_{\oplus}^{X^*}) \enspace .$$

Hence $X \in \min(\{ X, X^* \}, \leq_{\oplus}^{X})$. In other words, if $X^* \leq_{\oplus}^{X} X$, then it must also be the case that $X \leq_{\oplus}^{X} X^*$. Consequently, the order assignment generated by $\oplus$ is semi-faithful.

To show that it is also organised, consider well-defined sets of three-valued interpretations $\mathcal{M}, \mathcal{N}$ and three-valued interpretations X, Y such that

$$Y \in \min\left(\mathcal{M}, \leq_{\oplus}^{X}\right) \cup \min\left(\mathcal{M}, \leq_{\oplus}^{X^*}\right) \quad \text{and} \quad Y \in \min\left(\mathcal{N}, \leq_{\oplus}^{X}\right) \cup \min\left(\mathcal{N}, \leq_{\oplus}^{X^*}\right) \enspace .$$

By (D.9) we obtain that $Y \in [\![\|X\| \oplus \|\mathcal{M}\|]\!]_{\text{SE}}$ and $Y \in [\![\|X\| \oplus \|\mathcal{N}\|]\!]_{\text{SE}}$. Applying (P7)$_{\text{SE}}$ and (P4)$_{\text{SE}}$ yields that $Y \in [\![\|X\| \oplus \|\mathcal{M} \cup \mathcal{N}\|]\!]_{\text{SE}}$. Consequently, by (D.9), either $Y \in \min\left(\mathcal{M} \cup \mathcal{N}, \leq_{\oplus}^{X}\right)$ or $Y \in \min\left(\mathcal{M} \cup \mathcal{N}, \leq_{\oplus}^{X^*}\right)$, so the order assignment generated by $\oplus$ is organised. □

Lemma D.7. *Let ω be a semi-faithful preorder assignment and X a three-valued interpretation. Then there is no three-valued interpretation Y such that $Y <_{\omega}^{X} X$.*

Proof. We prove by contradiction. Suppose that $Y <_{\omega}^{X} X$ for some three-valued interpretation Y. Clearly, $Y \neq X$ due to irreflexivity of $<_{\omega}^{X}$ and $Y \neq X^*$ due to the second condition of semi-faithfulness. Hence $Y \neq X$ and $Y \neq X^*$, so by the first condition of semi-faithfulness, either $X <_{\omega}^{X} Y$ or $X^* <_{\omega}^{X} Y$. The former is in conflict with the irreflexivity of $<_{\omega}^{X}$ and in the latter case it follows by transitivity of $<_{\omega}^{X}$ that $X^* <_{\omega}^{X} X$, contrary to the second condition of semi-faithfulness. □

Proposition D.8. *Let $\oplus$ be a rule update operator. If $\oplus$ is characterised by a semi-faithful and organised preorder assignment, then it is also characterised by a faithful and organised partial order assignment.*

Proof. Let $\oplus$ be characterised by a semi-faithful and organised preorder assignment ω. We define the assignment ω' over $\mathfrak{X}$ as follows:

$$Y \leq_{\omega'}^{X} Z \quad \text{if and only if} \quad Y = X \vee Y = Z \vee Y <_{\omega}^{X} Z \enspace .$$

We need to show that $\leq_{\omega'}^{X}$ is a partial order for all $X \in \mathfrak{X}$, that ω' is faithful and organised and that for all programs $\mathcal{P}, \mathcal{U}$,

$$[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\text{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\text{SE}}} \min\left([\![\mathcal{U}]\!]_{\text{SE}}, \leq_{\omega'}^{X}\right) \enspace .$$

Note that due to Lemma D.7, the following holds for all three-valued interpretations X, Y:

$$\text{If } Y \leq_{\omega'}^{X} X \text{, then } Y = X. \tag{D.12}$$

Otherwise we would obtain that $Y <_\omega^X X$ which is in conflict with Lemma D.7.

Turning back to the main proof, reflexivity of $\leq_{\omega'}^X$ follows directly by its definition.

To show that $\leq_{\omega'}^X$ is antisymmetric, take some three-valued interpretations Y_1, Y_2 such that $Y_1 \leq_{\omega'}^X Y_2$ and $Y_2 \leq_{\omega'}^X Y_1$. If $Y_1 = X$, then $Y_2 \leq_{\omega'}^X X$ and it follows from (D.12) that $Y_2 = X = Y_1$. The case when $Y_2 = X$ is symmetric. If $Y_1 \neq X$ and $Y_2 \neq X$, then, by the definition of $\leq_{\omega'}^X$, either $Y_1 = Y_2$ as desired, or $Y_1 <_\omega^X Y_2$ and $Y_2 <_\omega^X Y_1$, which is in conflict with the transitivity and irreflexivity of $<_\omega^X$.

Turning to transitivity of $\leq_{\omega'}^X$, suppose that $Y_1 \leq_{\omega'}^X Y_2$ and $Y_2 \leq_{\omega'}^X Y_3$. We need to show that $Y_1 \leq_{\omega'}^X Y_3$. We consider three cases:

a) If $Y_1 = X$, then $Y_1 \leq_{\omega'}^X Y_3$ by the definition of $\leq_{\omega'}^X$.
b) If $Y_2 = X$, then $Y_1 \leq_{\omega'}^X X$, so $Y_1 = X$ due to (D.12) and the previous case applies.
c) If $Y_1 \neq X$ and $Y_2 \neq X$, then the desired conclusion follows from the transitivity of equality and of $<_\omega^X$.

As for faithfulness of ω', suppose that $Y \neq X$. We have $X \leq_{\omega'}^X Y$ by definition and $Y \not\leq_{\omega'}^X X$ follows from (D.12).

To show that ω' is organised, we prove the following property: For any well-defined set of three-valued interpretations $\mathcal{M}$ and any three-valued interpretation X,

$$\min\left(\mathcal{M}, \leq_{\omega'}^X\right) \cup \min\left(\mathcal{M}, \leq_{\omega'}^{X^*}\right) = \min\left(\mathcal{M}, \leq_{\omega}^X\right) \cup \min\left(\mathcal{M}, \leq_{\omega}^{X^*}\right) \quad . \tag{D.13}$$

From (D.13) it follows that since ω is organised, ω' must also be.

Before we prove (D.13), we need to note that $Y <_{\omega'}^X Z$ holds if and only if $Y \leq_{\omega'}^X Z$ and $Z \not\leq_{\omega'}^X Y$, so according to the definition of $\leq_{\omega'}^X$

$$Y <_{\omega'}^X Z \quad \text{if and only if} \quad (Y = X \vee Y = Z \vee Y <_\omega^X Z) \wedge (Z \neq X \wedge Z \neq Y \wedge Z \not<_\omega^X Y) \quad .$$

Due to Lemma D.7 and the transitivity and irreflexivity of $<_\omega^X$, this can be simplified to

$$Y <_{\omega'}^X Z \quad \text{if and only if} \quad (Y = X \wedge Y \neq Z) \vee Y <_\omega^X Z \quad . \tag{D.14}$$

Coming back to the proof of (D.13), we need to consider three cases:

a) If $X \notin \mathcal{M}$ and $X^* \notin \mathcal{M}$, then for all $Y, Z \in \mathcal{M}$, $Y \neq X$ and $Y \neq X^*$, so by (D.14),

$$Y <_{\omega'}^X Z \text{ if and only if } Y <_\omega^X Z \qquad \text{and} \qquad Y <_{\omega'}^{X^*} Z \text{ if and only if } Y <_\omega^{X^*} Z \quad ,$$

from which the desired conclusion follows directly.

b) If $X \notin \mathcal{M}$ and $X^* \in \mathcal{M}$, then for all $Y, Z \in \mathcal{M}$, $Y \neq X$, so by (D.14),

$$Y <_{\omega'}^X Z \text{ if and only if } Y <_\omega^X Z \quad .$$

Consequently, $\min(\mathcal{M}, \leq_{\omega'}^X) = \min(\mathcal{M}, \leq_\omega^X)$, and by (D.14) and semi-faithfulness of ω we obtain $\min(\mathcal{M}, \leq_{\omega'}^{X^*}) = \{ X^* \} = \min(\mathcal{M}, \leq_\omega^{X^*})$.

c) If $X \in \mathcal{M}$, then $X^* \in \mathcal{M}$, and by (D.14) and semi-faithfulness of ω,

$$\{ X \} \subseteq \min\left(\mathcal{M}, \leq_\omega^X\right) \subseteq \{ X, X^* \} \quad , \qquad \min\left(\mathcal{M}, \leq_\omega^{X^*}\right) = \{ X^* \} \quad ,$$
$$\min\left(\mathcal{M}, \leq_{\omega'}^X\right) = \{ X \} \quad , \qquad \min\left(\mathcal{M}, \leq_{\omega'}^{X^*}\right) = \{ X^* \} \quad ,$$

from which the desired conclusion follows straightforwardly.

Finally, it follows from the assumption that ω characterises $\oplus$ and from (D.13) that

$$\begin{aligned}
[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} &= \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}\right) \\
&= \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \left(\min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}\right) \cup \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X^*}\right)\right) \\
&= \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \left(\min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega'}^{X}\right) \cup \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega'}^{X^*}\right)\right) \\
&= \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega'}^{X}\right) \quad .
\end{aligned}$$

□

Proposition D.9. *Let $\oplus$ be a rule update operator. If $\oplus$ is characterised by a faithful and organised partial order assignment, then $\oplus$ satisfies conditions (P1)$_{\mathsf{SE}}$ – (P8)$_{\mathsf{SE}}$.*

Proof. Let $\oplus$ be characterised by a faithful and organised partial order assignment ω. We consider each condition separately:

(P1)$_{\mathsf{SE}}$ Since ω characterises $\oplus$, for all programs $\mathcal{P}, \mathcal{U}$,

$$[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}\right) \quad ,$$

so all elements of $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}}$ belong to $[\![\mathcal{U}]\!]_{\mathsf{SE}}$. Equivalently, $\mathcal{P} \oplus \mathcal{U} \models_{\mathsf{SE}} \mathcal{U}$.

(P2)$_{\mathsf{SE}}$ Suppose that $\mathcal{P} \models_{\mathsf{SE}} \mathcal{U}$ and take some $X \in [\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{U}]\!]_{\mathsf{SE}}$. Since the preorder assignment is faithful, for all $Y \in [\![\mathcal{U}]\!]_{\mathsf{SE}}$ with $Y \neq X$ we have $X <_{\omega}^{X} Y$. Consequently, $\min([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}) = \{ X \}$ and so

$$[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}\right) = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \{ X \} = [\![\mathcal{P}]\!]_{\mathsf{SE}} \quad .$$

(P3)$_{\mathsf{SE}}$ Suppose that $[\![\mathcal{P}]\!]_{\mathsf{SE}} \neq \emptyset$ and $[\![\mathcal{U}]\!]_{\mathsf{SE}} \neq \emptyset$. Then there is some $X_0 \in [\![\mathcal{P}]\!]_{\mathsf{SE}}$ and also some $Y \in \min([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X_0})$, so we obtain

$$Y \in \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X_0}\right) \subseteq \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}\right) = [\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} \quad .$$

Hence, $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} \neq \emptyset$.

(P4)$_{\mathsf{SE}}$ If $\mathcal{P} \equiv_{\mathsf{SE}} \mathcal{Q}$ and $\mathcal{U} \equiv_{\mathsf{SE}} \mathcal{V}$, then

$$\begin{aligned}
[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} &= \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}\right) = \bigcup_{X \in [\![\mathcal{Q}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{V}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X}\right) \\
&= [\![\mathcal{Q} \oplus \mathcal{V}]\!]_{\mathsf{SE}} \quad .
\end{aligned}$$

Therefore, $\mathcal{P} \oplus \mathcal{U} \equiv_{\mathsf{SE}} \mathcal{Q} \oplus \mathcal{V}$.

(P5)$_{\mathsf{SE}}$ Suppose that Y is an SE-model of $(\mathcal{P} \oplus \mathcal{U}) \dot{\wedge} \mathcal{V}$. Then $Y \in [\![\mathcal{V}]\!]_{\mathsf{SE}}$ and there is some SE-model X of $\mathcal{P}$ such that Y belongs to $\min([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X})$. Consequently, Y also belongs to $\min([\![\mathcal{U}]\!]_{\mathsf{SE}} \cap [\![\mathcal{V}]\!]_{\mathsf{SE}}, \leq_{\omega}^{X})$, so Y is an SE-model of $\mathcal{P} \oplus (\mathcal{U} \dot{\wedge} \mathcal{V})$.

(P6)$_{\mathsf{SE}}$ Assume that $\mathcal{P} \oplus \mathcal{U} \models_{\mathsf{SE}} \mathcal{V}$ and $\mathcal{P} \oplus \mathcal{V} \models_{\mathsf{SE}} \mathcal{U}$. We will prove by contradiction that $\mathcal{P} \oplus \mathcal{U} \models_{\mathsf{SE}} \mathcal{P} \oplus \mathcal{V}$. The other half can be proved similarly.
So suppose that Y is an SE-model of $\mathcal{P} \oplus \mathcal{U}$ but not of $\mathcal{P} \oplus \mathcal{V}$. Then there is some

SE-model X of $\mathcal{P}$ such that

$$Y \in \min(\llbracket \mathcal{U} \rrbracket_{\mathsf{SE}}, \leq_\omega^X) \enspace . \tag{D.15}$$

At the same time, there must be some SE-model Z of $\mathcal{V}$ such that $Z <_\omega^X Y$. Let Z_0 be minimal w.r.t. $\leq_\omega^X$ among all such Z. Then by transitivity of $<_\omega^X$ we obtain that $Z_0 \in \min(\llbracket \mathcal{V} \rrbracket_{\mathsf{SE}}, \leq_\omega^X)$ and, consequently, Z_0 is an SE-model of $\mathcal{P} \oplus \mathcal{V}$. By the assumption we now obtain that Z_0 is an SE-model of $\mathcal{U}$. But since $Z_0 <_\omega^X Y$, this is in conflict with (D.15).

(P7)$_{\mathsf{SE}}$ Suppose that $\mathcal{P}$ is strongly equivalent to $\|X\|$ for some three-valued interpretation X and Y is an SE-model of both $\mathcal{P} \oplus \mathcal{U}$ and $\mathcal{P} \oplus \mathcal{V}$. We will show that Y is an SE-model of $\mathcal{P} \oplus (\mathcal{U} \dot\vee \mathcal{V})$. Let $\mathcal{M} = \llbracket \mathcal{U} \rrbracket_{\mathsf{SE}}$ and $\mathcal{N} = \llbracket \mathcal{V} \rrbracket_{\mathsf{SE}}$. It follows that

$$Y \in \min(\mathcal{M}, \leq_\omega^X) \cup \min(\mathcal{M}, \leq_\omega^{X^*}) \text{ and } Y \in \min(\mathcal{N}, \leq_\omega^X) \cup \min(\mathcal{N}, \leq_\omega^{X^*}) \enspace ,$$

so since ω is organised, $Y \in \min(\mathcal{M} \cup \mathcal{N}, \leq_\omega^X) \cup \min(\mathcal{M} \cup \mathcal{N}, \leq_\omega^{X^*})$. Consequently, Y is an SE-model of $\mathcal{P} \oplus (\mathcal{U} \dot\vee \mathcal{V})$.

(P8)$_{\mathsf{SE}}$ The following sequence of equations establishes the property:

$$\begin{aligned}
\llbracket (\mathcal{P} \dot\vee \mathcal{Q}) \oplus \mathcal{U} \rrbracket_{\mathsf{SE}} &= \bigcup_{X \in \llbracket \mathcal{P} \dot\vee \mathcal{Q} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathcal{U} \rrbracket_{\mathsf{SE}}, \leq_\omega^X\right) \\
&= \bigcup_{X \in \llbracket \mathcal{P} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathcal{U} \rrbracket_{\mathsf{SE}}, \leq_\omega^X\right) \cup \bigcup_{X \in \llbracket \mathcal{Q} \rrbracket_{\mathsf{SE}}} \min\left(\llbracket \mathcal{U} \rrbracket_{\mathsf{SE}}, \leq_\omega^X\right) \\
&= \llbracket \mathcal{P} \oplus \mathcal{U} \rrbracket_{\mathsf{SE}} \cup \llbracket \mathcal{Q} \oplus \mathcal{U} \rrbracket_{\mathsf{SE}} \\
&= \llbracket (\mathcal{P} \oplus \mathcal{U}) \dot\vee (\mathcal{Q} \oplus \mathcal{U}) \rrbracket_{\mathsf{SE}}
\end{aligned}$$ □

Theorem 6.14. *Let $\oplus$ be a rule update operator. The following conditions are equivalent:*

a) The operator $\oplus$ satisfies conditions (P1)$_{\mathsf{SE}}$ – (P8)$_{\mathsf{SE}}$.

b) The operator $\oplus$ is characterised by a semi-faithful and organised preorder assignment.

c) The operator $\oplus$ is characterised by a faithful and organised partial order assignment.

Proof. Follows from Propositions D.6, D.8 and D.9. □

D.2 Specific Order Assignment for Rule Updates

Proposition D.10. *The assignment* W *is a preorder assignment.*

Proof. Recall that the assignment W is defined for all three-valued interpretations $X = (I, J)$, $Y = (K_1, L_1)$, $Z = (K_2, L_2)$ as follows: $Y \leq_{\mathsf{W}}^X Z$ if and only if

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

In order to show that W is a preorder assignment, we need to prove that given an arbitrary three-valued interpretation $X = (I, J)$, $\leq_{\mathsf{W}}^X$ is a preorder over $\mathfrak{X}$. This holds if and only if $\leq_{\mathsf{W}}^X$ is reflexive and transitive. First we show reflexivity. Take some three-valued interpretation $Y = (K, L)$. By definition $Y \leq_{\mathsf{W}}^X Y$ holds if and only if

1. $(L \div J) \subseteq (L \div J)$;
2. If $(L \div J) = (L \div J)$, then $(K \div I) \setminus \Delta \subseteq (K \div I) \setminus \Delta$ where $\Delta = L \div J$.

It is not difficult to check that both conditions hold.

To show transitivity, take some three-valued interpretations $Y_1 = (K_1, L_1), Y_2 = (K_2, L_2), Y_3 = (K_3, L_3)$ such that $Y_1 \leq_{\mathsf{W}}^{X} Y_2$ and $Y_2 \leq_{\mathsf{W}}^{X} Y_3$. We need to show that $Y_1 \leq_{\mathsf{W}}^{X} Y_3$. According to the definition of $\leq_{\mathsf{W}}^{X}$ we obtain

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$;

and also

1′ $(L_2 \div J) \subseteq (L_3 \div J)$;

2′ If $(L_2 \div J) = (L_3 \div J)$, then $(K_2 \div I) \setminus \Delta \subseteq (K_3 \div I) \setminus \Delta$ where $\Delta = L_2 \div J$.

We need to show the following two conditions:

1* $(L_1 \div J) \subseteq (L_3 \div J)$;

2* If $(L_1 \div J) = (L_3 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_3 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

It can be seen that 1* follows from 1. and 1′ by transitivity of the subset relation. To show that 2* holds as well, suppose that $(L_1 \div J) = (L_3 \div J)$. Then by 1. and 1′ we obtain that $(L_1 \div J) = (L_2 \div J) = (L_3 \div J) = \Delta$ and so by 2. and 2′ it holds that

$$(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta \subseteq (K_3 \div I) \setminus \Delta \ .$$

Consequently, 2* is also satisfied and our proof is finished. □

Lemma D.11. *Let $X = (I, J)$, $Y = (K_1, L_1)$, $Z = (K_2, L_2)$ be three-valued interpretations. Then $Y <_{\mathsf{W}}^{X} Z$ holds if and only if one of the following conditions is satisfied:*

a) $(L_1 \div J) \subsetneq (L_2 \div J)$*, or*

b) $(L_1 \div J) = (L_2 \div J)$ *and* $(K_1 \div I) \setminus \Delta \subsetneq (K_2 \div I) \setminus \Delta$ *where* $\Delta = L_1 \div J$*.*

Proof. By definition $Y <_{\mathsf{W}}^{X} Z$ holds if and only if $Y \leq_{\mathsf{W}}^{X} Z$ and not $Z \leq_{\mathsf{W}}^{X} Y$. This in turn holds if and only if the following two conditions hold

1. $(L_1 \div J) \subseteq (L_2 \div J)$;
2. If $(L_1 \div J) = (L_2 \div J)$, then $(K_1 \div I) \setminus \Delta \subseteq (K_2 \div I) \setminus \Delta$ where $\Delta = L_1 \div J$.

and one of the following conditions also holds:

i) $(L_2 \div J) \nsubseteq (L_1 \div J)$, or

ii) $(L_2 \div J) = (L_1 \div J)$ and $(K_2 \div I) \setminus \Delta \nsubseteq (K_1 \div I) \setminus \Delta$ where $\Delta = L_2 \div J$.

It is not difficult to verify that conditions 1., 2. and i) are together equivalent to a) and that conditions 1., 2. and ii) are together equivalent to b). This concludes our proof. □

Proposition D.12. *The assignment* W *is well-defined.*

Proof. By definition we need to show that there is a rule update operator $\oplus$ such that for all programs $\mathcal{P}, \mathcal{U}$,

$$[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{SE}} = \bigcup_{X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}} \min\left([\![\mathcal{U}]\!]_{\mathsf{SE}}, \leq_{\mathsf{W}}^{X}\right) \ .$$

This holds if and only if for every well-defined set of three-valued interpretations $\mathcal{M}$ and every three-valued interpretation X, the set of three-valued interpretations

$$\min\left(\mathcal{M}, \leq_{\mathsf{W}}^{X}\right) \cup \min\left(\mathcal{M}, \leq_{\mathsf{W}}^{X^*}\right) \tag{D.16}$$

is well-defined. Suppose that Y belongs to (D.16). We need to demonstrate that Y^* also belongs to (D.16). We consider two cases:

(a) Suppose that $Y \in \min(\mathcal{M}, \leq_{\mathsf{W}}^{X})$. If Y^* belongs to $\min(\mathcal{M}, \leq_{\mathsf{W}}^{X^*})$, then we are finished. On the other hand, if Y^* does not belong to $\min(\mathcal{M}, \leq_{\mathsf{W}}^{X^*})$, then there must be some $Z \in \mathcal{M}$ such that $Z <_{\mathsf{W}}^{X^*} Y^*$. Let $Y = (K_1, L_1), Z = (K_2, L_2)$ and $X = (I, J)$. By Lemma D.11 we know that $Z <_{\mathsf{W}}^{X^*} Y^*$ holds if and only if one of the following conditions is satisfied:

   a) $(L_2 \div J) \subsetneq (L_1 \div J)$, or

   b) $(L_2 \div J) = (L_1 \div J)$ and $(K_2 \div J) \setminus \Delta \subsetneq (L_1 \div J) \setminus \Delta$ where $\Delta = L_2 \div J$.

   If a) is satisfied, then Lemma D.11 implies that $Z <_{\mathsf{W}}^{X} Y$ which is in conflict with the assumption that $Y \in \min(\mathcal{M}, \leq_{\mathsf{W}}^{X})$. So b) must hold. But in that case we infer that $(K_2 \div J) \setminus \Delta$ is a proper subset of

$$(L_1 \div J) \setminus \Delta = (L_1 \div J) \setminus (L_1 \div J) = \emptyset \ ,$$

   which is impossible.

(b) Suppose that $Y \in \min(\mathcal{M}, \leq_{\mathsf{W}}^{X^*})$ and let $X = (I, J)$, $Y = (K, L)$. We first show that $Y^* \leq_{\mathsf{W}}^{X^*} Y$ holds – for this, the following conditions need to be satisfied:

   1. $(L \div J) \subseteq (L \div J)$;

   2. If $(L \div J) = (L \div J)$, then $(L \div J) \setminus \Delta \subseteq (K \div J) \setminus \Delta$ where $\Delta = L \div J$.

   It is not difficult to verify that both conditions hold.
   Thus, since $Y^* \leq_{\mathsf{W}}^{X^*} Y$, there can be no $Z \in \mathcal{M}$ with $Z <_{\mathsf{W}}^{X^*} Y^*$ because by transitivity we would obtain $Z <_{\mathsf{W}}^{X^*} Y$ which would be in conflict with the assumption that $Y \in \min(\mathcal{M}, \leq_{\mathsf{W}}^{X^*})$. So $Y^* \in \min(\mathcal{M}, \leq_{\mathsf{W}}^{X^*})$ and our proof is finished. □

Proposition D.13. *The assignment* W *is faithful.*

Proof. Take some three-valued interpretations $X = (I, J)$, $Y = (K, L)$ such that $Y \neq X$. We need to show that $X <_{\mathsf{W}}^{X} Y$. By Lemma D.11 this holds if and only if one of the following conditions is satisfied:

a) $(J \div J) \subsetneq (L \div J)$, or

b) $(J \div J) = (L \div J)$ and $(I \div I) \setminus \Delta \subsetneq (K \div I) \setminus \Delta$ where $\Delta = J \div J$.

We consider two cases:

i) If $L \div J = \emptyset$, then $L = J$ and since $Y \neq X$, we conclude that $K \neq I$. Consequently, the second condition is satisfied because $I \div I = \emptyset$ and $K \div I$ is non-empty.

ii) If $L \div J \neq \emptyset$, then a) holds since $J \div J = \emptyset$. □

Proposition D.14. *The assignment* W *is organised.*

Proof. Recall that by definition W is organised if for all three-valued interpretations X, Y and all well-defined sets of three-valued interpretations $\mathcal{M}, \mathcal{N}$ the following condition is satisfied:

If $Y \in \min(\mathcal{M}, \leq_{\mathsf{W}}^{X}) \cup \min(\mathcal{M}, \leq_{\mathsf{W}}^{X^*})$ and $Y \in \min(\mathcal{N}, \leq_{\mathsf{W}}^{X}) \cup \min(\mathcal{N}, \leq_{\mathsf{W}}^{X^*})$,
then $Y \in \min(\mathcal{M} \cup \mathcal{N}, \leq_{\mathsf{W}}^{X}) \cup \min(\mathcal{M} \cup \mathcal{N}, \leq_{\mathsf{W}}^{X^*})$.

Suppose that $Y \notin \min(\mathcal{M} \cup \mathcal{N}, \leq_{\mathsf{W}}^{X}) \cup \min(\mathcal{M} \cup \mathcal{N}, \leq_{\mathsf{W}}^{X^*})$. We need to show that at least one of the following holds:

i) $Y \notin \min(\mathcal{M}, \leq_{\mathsf{W}}^{X}) \cup \min(\mathcal{M}, \leq_{\mathsf{W}}^{X^*})$;

ii) $Y \notin \min(\mathcal{N}, \leq_{\mathsf{W}}^{X}) \cup \min(\mathcal{N}, \leq_{\mathsf{W}}^{X^*})$.

If $Y \notin \mathcal{M}$, then i) is trivially satisfied. Similarly, if $Y \notin \mathcal{N}$, then ii) is trivially satisfied. So we can assume that $Y \in \mathcal{M} \cap \mathcal{N}$. Then it follows from the assumption that there must be some $Z_1, Z_2 \in \mathcal{M} \cup \mathcal{N}$ such that $Z_1 <_{\mathsf{W}}^{X} Y$ and $Z_2 <_{\mathsf{W}}^{X^*} Y$. If Z_1 and Z_2 both belong to $\mathcal{M}$, then i) is satisfied; if they both belong to $\mathcal{N}$, then ii) is satisfied. So let's assume, without loss of generality, that $Z_1 \in \mathcal{M}$ and $Z_2 \in \mathcal{N}$. Furthermore, let $X = (I, J)$, $Y = (K, L)$, $Z_1 = (K_1, L_1)$ and $Z_2 = (K_2, L_2)$. Then it follows from $Z_2 <_{\mathsf{W}}^{X^*} Y$ and Lemma D.11 that we need to consider two cases:

a) If $(L_2 \div J) \subsetneq (L \div J)$, then by Lemma D.11 we also have $Z_2 <_{\mathsf{W}}^{X} Y$ and, consequently, ii) is satisfied.

b) If $(L_2 \div J) = (L \div J)$ and $(K_2 \div J) \setminus \Delta \subsetneq (K \div J) \setminus \Delta$ where $\Delta = (L_2 \div J)$, then it follows that $(K \div J) \setminus \Delta \neq \emptyset$ and by using $\Delta = L_2 \div J = L \div J$ we obtain

$$(K \div J) \setminus (L \div J) \neq \emptyset \ . \tag{D.17}$$

Furthermore, from $Z_1 <_{\mathsf{W}}^{X} Y$ we know that one of the following cases occurs:

a') $(L_1 \div J) \subsetneq (L \div J)$, or

b') $(L_1 \div J) = (L \div J)$ and $(K_1 \div I) \setminus \Delta \subsetneq (K \div I) \setminus \Delta$, where $\Delta = L_1 \div J$.

We will show that $Z_1^* <_{\mathsf{W}}^{X^*} Y$. By Lemma D.11 this holds if and only if one of the following conditions is satisfied:

a*) $(L_1 \div J) \subsetneq (L \div J)$, or

b*) $(L_1 \div J) = (L \div J)$ and $(L_1 \div J) \setminus \Delta \subsetneq (K \div J) \setminus \Delta$, where $\Delta = L_1 \div J$.

We see that a') implies a*) and b') together with (D.17) implies b*). Also, since $\mathcal{M}$ is well-defined, we have $Z_1^* \in \mathcal{M}$, so i) is satisfied. □

Proposition 6.15. *The assignment* W *is a well-defined, faithful and organised preorder assignment.*

Proof. Follows by Propositions D.10, D.12, D.13 and D.14. □

E

Proofs: Semantic Characterisations of Rules and Programs

In the following we present proofs of results from Chapter 7, implicitly working under the same assumptions as those imposed in that chapter. That is, we constrain ourselves to propositional logic programs without explicit negation over a finite set of propositional atoms $\mathcal{A}$.

Definition E.1. Let π be a rule. By π^+ and π^-, respectively, we denote the rules

$$H(\pi)^+ \leftarrow B(\pi)^+. \qquad \text{and} \qquad {\sim}H(\pi)^- \leftarrow {\sim}B(\pi)^-.$$

Lemma E.2. *Let π be a rule and J an interpretation. Then,*

$$J \models \pi \qquad \textit{if and only if} \qquad J \models \pi^+ \lor J \models \pi^- \ .$$

Proof. Suppose first that $J \models \pi$. Then one of the following cases applies:

a) If $J \not\models L$ for some literal $L \in B(\pi)$, then, depending on whether L is an atom or a default literal, it follows that $J \models \pi^+$ or $J \models \pi^-$, respectively.

b) If $J \models L$ for some literal $L \in H(\pi)$, then, depending on whether L is an atom or a default literal, it follows that $J \models \pi^+$ or $J \models \pi^-$, respectively.

On the other hand, if $J \models \pi^+$, then either $J \not\models p$ for some atom $p \in B(\pi)$, or $J \models p$ for some atom $p \in H(\pi)$. In either case it follows that $J \models \pi$.

Similarly, if $J \models \pi^-$, then either $J \not\models L$ for some default literal $L \in B(\pi)$, or $J \models L$ for some default literal $L \in H(\pi)$. In either case it follows that $J \models \pi$. □

Lemma E.3. *Let π be a rule and J an interpretation. Then,*

$$\pi^J = \begin{cases} \tau & J \models \pi^- \ ; \\ \pi^+ & \textit{otherwise} \ . \end{cases}$$

Proof. Follows directly from the definitions of π^J, π^+ and π^-. □

E.1 SE-Models vs. Individual Rules

Lemma E.4. *Let (I, J) be a three-valued interpretation and π a rule. Then,*

$$(I, J) \in [\![\pi]\!]_{\mathsf{SE}} \quad \text{if and only if} \quad (I \models \pi^+ \wedge J \models \pi^+) \vee J \models \pi^- \enspace .$$

Proof. First take some SE-model (I, J) of π and suppose that $J \not\models \pi^-$. We need to prove that $I \models \pi^+$ and $J \models \pi^+$. It follows from the definition of SE-models that $J \models \pi$ and $I \models \pi^J$, so by Lemma E.2 we obtain $J \models \pi^+$. Furthermore, it follows from Lemma E.3 that $\pi^J = \pi^+$, so it also holds that $I \models \pi^+$.

Turning to the converse implication, first suppose that $J \models \pi^-$. By Lemmas E.2 and E.3 we obtain $J \models \pi$ and $\pi^J = \tau$, so $I \models \pi^J$ and thus (I, J) is an SE-model of π. If $J \not\models \pi^-$, then we need to show that $I \models \pi^+$ together with $J \models \pi^+$ imply that (I, J) is an SE-model of π. We obtain $J \models \pi$ from Lemma E.2 and $I \models \pi^J$ holds because $\pi^J = \pi^+$ by Lemma E.3. □

E.1.1 SE-Canonical Rules

Lemma 7.3. *Rules of the following forms are SE-tautological:*

$$p; H \leftarrow p, B. \qquad\qquad H; {\sim}p \leftarrow B, {\sim}p. \qquad\qquad H \leftarrow B, p, {\sim}p.$$

Proof. First assume that a rule π is of the first form and take some three-valued interpretation (I, J). We need to show that (I, J) is an SE-model of π. According to Lemma E.4, it suffices to show that $I \models \pi^+$ and $J \models \pi^+$. Both of these properties follow from the fact that p belongs to both $H(\pi)^+$ and $B(\pi)^+$.

Now suppose that π is of the second form. Given a three-valued interpretation (I, J), we see that the atom p belongs to both $H(\pi)^-$ and $B(\pi)^-$, so $J \models \pi^-$. Hence we can use Lemma E.4 to conclude that (I, J) is an SE-model of π.

Finally, suppose that π takes the third form and take some three-valued interpretation (I, J). If $J \models \pi^-$, then (I, J) is an SE-model of π by Lemma E.4. On the other hand, if $J \not\models \pi^-$, then $J \models {\sim}p$. Consequently, $I \not\models p$ because I is a subset of J. These findings imply that $I \models \pi^+$ and $J \models \pi^+$, so by using Lemma E.4 we can once again conclude that (I, J) is an SE-model of π. □

Lemma E.5. *Rules of the following forms are SE-equivalent:*

$$p; H \leftarrow B, {\sim}p. \qquad\qquad H \leftarrow B, {\sim}p.$$

Proof. Let the first rule be denoted by π_1, the second by π_2 and take some three-valued interpretation (I, J). We need to show that (I, J) is an SE-model of π_1 if and only if it is an SE-model of π_2. According to Lemma E.4, it suffices to prove the following:

$$(I \models \pi_1^+ \wedge J \models \pi_1^+) \vee J \models \pi_1^- \quad \text{if and only if} \quad (I \models \pi_2^+ \wedge J \models \pi_2^+) \vee J \models \pi_2^- \enspace . \tag{E.1}$$

Note first that $\pi_1^- = \pi_2^-$, so

$$J \models \pi_1^- \quad \text{if and only if} \quad J \models \pi_2^- \enspace . \tag{E.2}$$

Now consider two cases:

a) If $J \models \pi_1^-$, then $J \models \pi_2^-$ due to (E.2) and we can conclude that (E.1) holds.

b) If $J \not\models \pi_1^-$, then $J \not\models \pi_2^-$ due to (E.2) and (E.1) reduces to the following goal:

$$I \models \pi_1^+ \wedge J \models \pi_1^+ \quad \text{if and only if} \quad I \models \pi_2^+ \wedge J \models \pi_2^+ \ . \tag{E.3}$$

Now it suffices to observe that $J \not\models \pi_1^-$ implies $J \models {\sim}p$, hence $J \not\models p$, and since I is a subset of J, we can conclude that $I \not\models p$. Since π_1^+ differs from π_2^+ only in the head atom p, it follows that (E.3) is satisfied. □

Lemma E.6. *Rules of the following forms are SE-equivalent:*

$$H; {\sim}p \leftarrow p, B. \qquad\qquad H \leftarrow p, B.$$

Proof. Let the first rule be denoted by π_1, the second by π_2 and take some three-valued interpretation (I, J). We need to show that (I, J) is an SE-model of π_1 if and only if it is an SE-model of π_2. According to Lemma E.4, it suffices to prove the following:

$$(I \models \pi_1^+ \wedge J \models \pi_1^+) \vee J \models \pi_1^- \quad \text{if and only if} \quad (I \models \pi_2^+ \wedge J \models \pi_2^+) \vee J \models \pi_2^- \ . \tag{E.4}$$

Note first that $\pi_1^+ = \pi_2^+$, so

$$I \models \pi_1^+ \wedge J \models \pi_1^+ \quad \text{if and only if} \quad I \models \pi_2^+ \wedge J \models \pi_2^+ \ . \tag{E.5}$$

Now consider two cases:

a) If both $I \models \pi_1^+$ and $J \models \pi_1^+$, then $I \models \pi_2^+$ and $J \models \pi_2^+$ due to (E.5) and we can conclude that (E.4) holds.

b) If either $I \not\models \pi_1^+$ or $J \not\models \pi_1^+$, then $I \not\models \pi_2^+$ or $J \not\models \pi_2^+$ due to (E.5) and (E.4) reduces to the following goal:

$$J \models \pi_1^- \quad \text{if and only if} \quad J \models \pi_2^- \ . \tag{E.6}$$

Now it suffices to observe that $I \not\models \pi_1^+$ implies $I \models p$ and since I is a subset of J, we can conclude that $J \not\models {\sim}p$. Similarly, $J \not\models \pi_1^+$ implies $J \models p$ so that $J \not\models {\sim}p$. Since π_1^- differs from π_2^- only in the head literal ${\sim}p$, it follows that (E.6) is satisfied. □

Lemma 7.4. *Rules of the following forms are SE-equivalent:*

$$H; {\sim}L \leftarrow L, B. \qquad\qquad H \leftarrow L, B.$$

Proof. Follows from Lemmas E.5 and E.6. □

Lemma 7.5. *Rules of the following forms are SE-equivalent:*

$${\sim}p; {\sim}H^- \leftarrow B. \qquad\qquad {\sim}H^- \leftarrow p, B.$$

Proof. Let the first rule be denoted by π_1, the second by π_2 and take some three-valued interpretation (I, J). We need to prove that (I, J) is an SE-model of π_1 if and only if it is an SE-model of π_2. According to Lemma E.4, it suffices to prove the following:

$$(I \models \pi_1^+ \wedge J \models \pi_1^+) \vee J \models \pi_1^- \quad \text{if and only if} \quad (I \models \pi_2^+ \wedge J \models \pi_2^+) \vee J \models \pi_2^- \ . \tag{E.7}$$

Note that the rules π_1^+, π_1^-, π_2^+ and π_2^- are as follows:

$$\begin{array}{llll} \pi_1^+ : & \leftarrow B^+. & \pi_1^- : & \sim p; \sim H^- \leftarrow \sim B^-. \\ \pi_2^+ : & \leftarrow p, B^+. & \pi_2^- : & \sim H^- \leftarrow \sim B^-. \end{array}$$

Suppose first that the left-hand side of (E.7) is satisfied. If $I \models \pi_1^+$ and $J \models \pi_1^+$, then clearly also $I \models \pi_2^+$ and $J \models \pi_2^+$, i.e. the right-hand side is also satisfied. If $J \models \pi_1^-$, then either $J \models \pi_2^-$, which also satisfies the right-hand side, or $J \models \sim p$. In the latter case it follows that $J \not\models p$ and $I \not\models p$ because I is a subset of J. Consequently, $I \models \pi_2^+$ and $J \models \pi_2^+$, once again satisfying the right-hand side.

Now suppose that the right-hand side of (E.7) is satisfied. If $I \models \pi_2^+$ and $J \models \pi_2^+$, then either $J \models \pi_1^+$ and thus also $I \models \pi_1^+$, or $J \not\models p$, in which case it follows that $J \models \pi_1^-$. In either case, the left-hand side is satisfied. The case that remains is when $J \models \pi_2^-$, which directly entails that $J \models \pi_1^-$. □

Theorem 7.8. *Every rule π is SE-equivalent to the SE-canonical rule $\mathsf{can}_{\mathsf{SE}}(\pi)$.*

Proof. This can be shown by a careful iterative application of Lemmas 7.3, E.5, E.6 and 7.5. First observe that if $\mathsf{can}_{\mathsf{SE}}(\pi) = \tau$, then Lemma 7.3 implies that π is SE-tautological, thus indeed SE-equivalent to τ.

In the principal case we can use Lemma E.5 on all atoms from $H(\pi)^+ \cap B(\pi)^-$ and remove them one by one from $H(\pi)^+$ while preserving SE-models. A similar situation occurs with atoms from $H(\pi)^- \cap B(\pi)^+$ which can be, according to Lemma E.6, removed from $H(\pi)^-$ without affecting SE-models. After these steps are performed, we obtain the rule

$$(H(\pi)^+ \setminus B(\pi)^-); \sim(H(\pi)^- \setminus B(\pi)^+) \leftarrow B(\pi)^+, \sim B(\pi)^-. \tag{E.8}$$

This is also the result of $\mathsf{can}_{\mathsf{SE}}(\pi)$, unless the set $H(\pi)^+ \setminus B(\pi)^-$ is empty. In that case, one can repeatedly apply Lemma 7.5 and move the atoms from the negative head of rule (E.8) into its positive body. This way, we obtain the rule

$$\leftarrow (B(\pi)^+ \cup H(\pi)^-), \sim B(\pi)^-.$$

which also coincides with the result of $\mathsf{can}_{\mathsf{SE}}(\pi)$. □

E.1.2 Reconstructing Rules from SE-Models

Lemma 7.9. *Let (I, J) be a three-valued interpretation and π a rule. Then $(I, J) \notin [\![\pi]\!]_{\mathsf{SE}}$ if and only if the following conditions are satisfied:*

1. *Either $B(\pi)^+ \subseteq I \subseteq \mathcal{A} \setminus H(\pi)^+$ or $J \cap H(\pi)^+ = \emptyset$ and*
2. *$H(\pi)^- \cup B(\pi)^+ \subseteq J \subseteq \mathcal{A} \setminus B(\pi)^-$.*

Proof. Note first that since I is a subset of J, we can equivalently state the conditions above as follows:

1′ Either $B(\pi)^+ \subseteq I \subseteq \mathcal{A} \setminus H(\pi)^+$ or $B(\pi)^+ \subseteq J \subseteq \mathcal{A} \setminus H(\pi)^+$ and

2′ $H(\pi)^- \subseteq J \subseteq \mathcal{A} \setminus B(\pi)^-$.

By Lemma E.4, $(I, J) \notin [\![\pi]\!]_{\mathsf{SE}}$ if and only if $(I \not\models \pi^+ \vee J \not\models \pi^+) \wedge J \not\models \pi^-$. We will show that this is equivalent to the two conditions above.

We know that $I \not\models \pi^+$ if and only if

$$\forall p \in B(\pi)^+ : I \models p \qquad \text{and} \qquad \forall p \in H(\pi)^+ : I \not\models p \enspace ,$$

which is equivalent to

$$\forall p \in B(\pi)^+ : p \in I \qquad \text{and} \qquad \forall p \in H(\pi)^+ : p \notin I \enspace .$$

In other words, $I \not\models \pi^+$ if and only if $B(\pi)^+ \subseteq I$ and $I \cap H(\pi)^+ = \emptyset$. Similarly, $J \not\models \pi^+$ if and only $B(\pi)^+ \subseteq J$ and $J \cap H(\pi)^+ = \emptyset$. Consequently, $I \not\models \pi^+ \vee J \not\models \pi^+$ if and only if condition 1′ holds.

On the other hand, $J \not\models \pi^-$ holds if and only if

$$\forall {\sim}p \in {\sim}B(\pi)^- : J \models {\sim}p \qquad \text{and} \qquad \forall {\sim}p \in {\sim}H(\pi)^- : J \not\models {\sim}p \enspace ,$$

which is equivalent to

$$\forall p \in B(\pi)^- : p \notin J \qquad \text{and} \qquad \forall p \in H(\pi)^- : p \in J \enspace .$$

In other words, $J \cap B(\pi)^- = \emptyset$ and $H(\pi)^- \subseteq J$, equivalently to condition 2′ above. □

Corollary E.7. *Let π be an SE-canonical rule different from τ, put $I = B(\pi)^+$, $J = H(\pi)^- \cup B(\pi)^+$ and $J' = \mathcal{A} \setminus B(\pi)^-$, and let p be an atom. Then the following holds:*

(1) $(I, J) \notin [\![\pi]\!]_{\mathsf{SE}}$.

(2) $(I, J \cup \{ p \}) \in [\![\pi]\!]_{\mathsf{SE}}$ *if and only if* $p \in B(\pi)^-$.

(3) $(I \cup \{ p \}, J \cup \{ p \}) \in [\![\pi]\!]_{\mathsf{SE}}$ *if and only if* $p \in H(\pi)^+ \cup B(\pi)^-$.

(4) $(I, J') \notin [\![\pi]\!]_{\mathsf{SE}}$.

Proof. All parts of the corollary follow from Lemma 7.9 and the disjointness properties satisfied by SE-canonical rules. □

Lemma E.8. *Let π be an SE-canonical rule different from τ and p an atom. Then $p \in B(\pi)^-$ if and only if for all $(I, J) \in \mathcal{X}$,*

$$p \in J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace .$$

Proof. Suppose that $p \in B(\pi)^-$ and take some three-valued interpretation (I, J) with $p \in J$. Then $J \not\models {\sim}p$, so it follows that $J \models \pi^-$. Consequently, by Lemma E.4, $(I, J) \in [\![\pi]\!]_{\mathsf{SE}}$.

To prove the converse implication, let $I = B(\pi)^+$ and $J = H(\pi)^- \cup B(\pi)^+$. It follows that $(I, J \cup \{ p \}) \in [\![\pi]\!]_{\mathsf{SE}}$, so by Corollary E.7(2) we conclude that $p \in B(\pi)^-$. □

Lemma E.9. *Let π be an SE-canonical rule different from τ and p an atom. Then $p \in H(\pi)^+$ if and only if $p \notin B(\pi)^-$ and for all $(I, J) \in \mathcal{X}$,*

$$p \in I \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace .$$

Proof. Suppose that $p \in H(\pi)^+$ and take some three-valued interpretation (I, J) with $p \in I$. Then $I \models p$ and $J \models p$, so it follows that $I \models \pi^+$ and $J \models \pi^+$. Consequently, by Lemma E.4, $(I, J) \in [\![\pi]\!]_{\mathsf{SE}}$.

To prove the converse implication, let $I = B(\pi)^+$ and $J = H(\pi)^- \cup B(\pi)^+$. It follows that $(I \cup \{p\}, J \cup \{p\}) \in [\![\pi]\!]_{\mathsf{SE}}$, so by Corollary E.7(3) we conclude that $p \in H(\pi)^+ \cup B(\pi)^-$. Moreover, by the assumption we know that $p \notin B(\pi)^-$, so $p \in H(\pi)^+$. □

Lemma 7.10. *Let π be an SE-canonical rule different from τ and p an atom. Then:*

- $p \in B(\pi)^-$ *if and only if for all* $(I, J) \in \mathcal{X}$,

$$p \in J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace ;$$

- $p \in H(\pi)^+$ *if and only if* $p \notin B(\pi)^-$ *and for all* $(I, J) \in \mathcal{X}$,

$$p \in I \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace .$$

Proof. Follows from Lemmas E.8 and E.9. □

Lemma E.10. *Let π be an SE-canonical rule different from τ and p an atom. Then $p \in B(\pi)^+$ if and only if for all $(I, J) \in \mathcal{X}$,*

$$(p \notin I \wedge J \cap H(\pi)^+ \neq \emptyset) \vee p \notin J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace .$$

Proof. Suppose that $p \in B(\pi)^+$ and take some three-valued interpretation (I, J). If $p \notin I$ and $J \cap H(\pi)^+ \neq \emptyset$, then $J \models \pi^+$ and $I \models \pi^+$. On the other hand, if $p \notin J$, then $p \notin I$ since I is a subset of J, and we can once again conclude that $J \models \pi^+$ and $I \models \pi^+$. Consequently, by Lemma E.4, $(I, J) \in [\![\pi]\!]_{\mathsf{SE}}$.

To prove the converse implication, we consider two cases:

a) If $H(\pi)^+ = \emptyset$, then $H(\pi)^- = \emptyset$ because π is SE-canonical. Put $I = J = B(\pi)^+$. According to Corollary E.7(1), $(I, J) \notin [\![\pi]\!]_{\mathsf{SE}}$ while $(I \setminus \{p\}, J \setminus \{p\}) \in [\![\pi]\!]_{\mathsf{SE}}$ by the assumption. Consequently, J must be different from $J \setminus \{p\}$, so $p \in J = B(\pi)^+$.

b) If $H(\pi)^+ \neq \emptyset$, then put $I = B(\pi)^+$ and $J' = \mathcal{A} \setminus B(\pi)^-$. We obtain $(I, J') \notin [\![\pi]\!]_{\mathsf{SE}}$ by Corollary E.7(4). Furthermore, $J' \cap H(\pi)^+ \neq \emptyset$ because, since π is SE-canonical, $H(\pi)^+ \cap B(\pi)^- = \emptyset$. It thus follows by the assumption that $(I \setminus \{p\}, J') \in [\![\pi]\!]_{\mathsf{SE}}$. Consequently, I must be different from $I \setminus \{p\}$, so $p \in I = B(\pi)^+$. □

Lemma E.11. *Let p be an atom. Then $p \in H(\pi)^-$ if and only if $p \notin B(\pi)^+$ and for all $(I, J) \in \mathcal{X}$,*

$$p \notin J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace .$$

Proof. Suppose that $p \in H(\pi)^-$ and take some three-valued interpretation (I, J) with $p \notin I$. Then $J \models \sim p$, so it follows that $J \models \pi^-$. Consequently, by Lemma E.4, $(I, J) \in [\![\pi]\!]_{\mathsf{SE}}$.

To prove the converse implication, let $I = B(\pi)^+$ and $J = H(\pi)^- \cup B(\pi)^+$. It follows by Corrolary E.7(1) that $(I, J) \notin [\![\pi]\!]_{\mathsf{SE}}$ while by our assumption, $(I \setminus \{p\}, J \setminus \{p\}) \in [\![\pi]\!]_{\mathsf{SE}}$. Consequently, J must differ from $J \setminus \{p\}$, which implies that $p \in J = H(\pi)^- \cup B(\pi)^+$. Furthermore, by the assumption $p \notin B(\pi)^+$, so we can conclude that $p \in H(\pi)^-$. □

Lemma 7.11. *Let p be an atom. Then:*

- $p \in B(\pi)^+$ *if and only if for all* $(I, J) \in \mathcal{X}$,

$$(p \notin I \wedge J \cap H(\pi)^+ \neq \emptyset) \vee p \notin J \textit{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace ;$$

- $p \in H(\pi)^-$ *if and only if* $p \notin B(\pi)^+$ *and for all* $(I, J) \in \mathcal{X}$,

$$p \notin J \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{SE}} \enspace .$$

Proof. Follows from Lemmas E.10 and E.11. □

Theorem 7.13. *For every SE-canonical rule* π, $\|[\![\pi]\!]_{\mathsf{SE}}\|_{\mathsf{SE}} = \pi$.

Proof. If $\pi = \tau$, then $[\![\pi]\!]_{\mathsf{SE}} = \mathcal{X}$ and, by Definition 7.12, $\|\mathcal{X}\|_{\mathsf{SE}} = \tau = \pi$, so the identity is satisfied. In the principal case, π is an SE-canonical rule different from τ. Let $\mathcal{M} = [\![\pi]\!]_{\mathsf{SE}}$. It follows from Definition 7.12 and Lemmas 7.10 and 7.11 that $\pi = \|\mathcal{M}\|_{\mathsf{SE}}$. □

E.1.3 SE-Rule-Expressible Sets of Interpretations

Lemma E.12. *Let* $\mathcal{M}$ *be a set of three-valued interpretations different from* $\mathcal{X}$*. Then the sets* $H_{\mathsf{SE}}(\mathcal{M})^+ \cup H_{\mathsf{SE}}(\mathcal{M})^-$, $B_{\mathsf{SE}}(\mathcal{M})^+$ *and* $B_{\mathsf{SE}}(\mathcal{M})^-$ *are pairwise disjoint.*

Proof. First suppose that p is a member of both $H_{\mathsf{SE}}(\mathcal{M})^+ \cup H_{\mathsf{SE}}(\mathcal{M})^-$ and $B_{\mathsf{SE}}(\mathcal{M})^+$. Since p belongs to $B_{\mathsf{SE}}(\mathcal{M})^+$, it cannot belong to $H_{\mathsf{SE}}(\mathcal{M})^-$ by definition. So p belongs to both $H_{\mathsf{SE}}(\mathcal{M})^+$ and $B_{\mathsf{SE}}(\mathcal{M})^+$. Take an arbitrary three-valued interpretation (I, J). If $p \in I$, then, by the definition of $H_{\mathsf{SE}}(\mathcal{M})^+$, $(I, J) \in \mathcal{M}$. If $p \notin I$ but $p \in J$, then $J \cap H_{\mathsf{SE}}(\mathcal{M})^+ \neq \emptyset$, so, by the definition of $B_{\mathsf{SE}}(\mathcal{M})^+$, $(I, J) \in \mathcal{M}$. If $p \notin J$, then, by the definition of $B_{\mathsf{SE}}(\mathcal{M})^+$, once again $(I, J) \in \mathcal{M}$. As a consequence, $\mathcal{M} = \mathcal{X}$, contrary to our assumption.

Now suppose that p is a member of both $H_{\mathsf{SE}}(\mathcal{M})^+ \cup H_{\mathsf{SE}}(\mathcal{M})^-$ and $B_{\mathsf{SE}}(\mathcal{M})^-$. Then, since p belongs to $B_{\mathsf{SE}}(\mathcal{M})^-$, it cannot belong to $H_{\mathsf{SE}}(\mathcal{M})^+$ by definition. So p belongs to both $H_{\mathsf{SE}}(\mathcal{M})^-$ and $B_{\mathsf{SE}}(\mathcal{M})^-$. Take an arbitrary three-valued interpretation (I, J). If $p \in J$, then, by the definition of $B_{\mathsf{SE}}(\mathcal{M})^-$, $(I, J) \in \mathcal{M}$. If $p \notin J$, then, by the definition of $H_{\mathsf{SE}}(\mathcal{M})^-$, $(I, J) \in \mathcal{M}$. As a consequence, $\mathcal{M} = \mathcal{X}$, contrary to our assumption.

Finally, suppose that p is a member of both $B_{\mathsf{SE}}(\mathcal{M})^+$ and $B_{\mathsf{SE}}(\mathcal{M})^-$ and take an arbitrary three-valued interpretation (I, J). If $p \in J$, then, by the definition of $B_{\mathsf{SE}}(\mathcal{M})^-$, $(I, J) \in \mathcal{M}$. If $p \notin J$, then, by the definition of $B_{\mathsf{SE}}(\mathcal{M})^+$, we obtain $(I, J) \in \mathcal{M}$ once again. Consequently, $\mathcal{M} = \mathcal{X}$, contrary to our assumption. □

Lemma E.13. *For every set of three-valued interpretations* $\mathcal{M}$, $\|\mathcal{M}\|_{\mathsf{SE}}$ *is an SE-canonical rule.*

Proof. If $\mathcal{M} = \mathcal{X}$, then $\|\mathcal{M}\|_{\mathsf{SE}} = \tau$ and the proof is finished. Otherwise, $\|\mathcal{M}\|_{\mathsf{SE}}$ is of the form

$$H_{\mathsf{SE}}(\mathcal{M})^+; {\sim}H_{\mathsf{SE}}(\mathcal{M})^- \leftarrow B_{\mathsf{SE}}(\mathcal{M})^+, {\sim}B_{\mathsf{SE}}(\mathcal{M})^- .$$

To show that this rule is SE-canonical, we need to prove that the sets $H_{\mathsf{SE}}(\mathcal{M})^+ \cup H_{\mathsf{SE}}(\mathcal{M})^-$, $B_{\mathsf{SE}}(\mathcal{M})^+$ and $B_{\mathsf{SE}}(\mathcal{M})^-$ are pairwise disjoint and that $H_{\mathsf{SE}}(\mathcal{M})^- = \emptyset$ whenever $H_{\mathsf{SE}}(\mathcal{M})^+ = \emptyset$. The former follows from Lemma E.34. As for the latter, suppose that $H_{\mathsf{SE}}(\mathcal{M})^+ = \emptyset$. Then $J \cap H_{\mathsf{SE}}(\mathcal{M})^+ = \emptyset$ for all $J \in \mathcal{I}$, so it follows from the definitions of $B_{\mathsf{SE}}(\mathcal{M})^+$ and $H_{\mathsf{SE}}(\mathcal{M})^-$ that $H_{\mathsf{SE}}(\mathcal{M})^- = \emptyset$. □

Lemma 7.17. *The set of all SE-models of an SE-canonical rule* π *is the least among all sets of three-valued interpretations* $\mathcal{M}$ *such that* $\|\mathcal{M}\|_{\mathsf{SE}} = \pi$.

Proof. Let π be an SE-canonical rule. From Theorem 7.13 we know that $\|[\![\pi]\!]_{\mathsf{SE}}\|_{\mathsf{SE}} = \pi$, so it remains to show that $[\![\pi]\!]_{\mathsf{SE}}$ is a subset of every set of interpretations $\mathcal{M}$ such that $\|\mathcal{M}\|_{\mathsf{SE}} = \pi$. Take one such $\mathcal{M}$. If $\pi = \tau$, then $[\![\pi]\!]_{\mathsf{SE}} = \mathcal{X}$. Furthermore, $\mathcal{M} = \mathcal{X}$, for otherwise the set $H_{\mathsf{SE}}(\mathcal{M})^+ \cap B_{\mathsf{SE}}(\mathcal{M})^+$ would be non-empty, contrary to Lemma E.12.

In the principal case, π is not the canonical tautology, so from $\|\mathcal{M}\|_{\mathsf{SE}} = \pi$ we obtain that $\mathcal{M} \neq \mathfrak{X}$ as well as $H(\pi)^+ = H_{\mathsf{SE}}(\mathcal{M})^+$, $H(\pi)^- = H_{\mathsf{SE}}(\mathcal{M})^-$, $B(\pi)^+ = B_{\mathsf{SE}}(\mathcal{M})^+$ and $B(\pi)^- = B_{\mathsf{SE}}(\mathcal{M})^-$. Take some $(I, J) \in [\![\pi]\!]_{\mathsf{SE}}$, we need to show that $(I, J) \in \mathcal{M}$. By Lemma E.4 we obtain that either $I \models \pi^+$ and $J \models \pi^+$, or $J \models \pi^-$. This means that we have five cases to consider:

a) If $I \models p$ for some $p \in H(\pi)^+$, then $p \in H_{\mathsf{SE}}(\mathcal{M})^+$ and $p \in I$, so it follows from the definition of $H_{\mathsf{SE}}(\mathcal{M})^+$ that $(I, J) \in \mathcal{M}$.

b) If $I \not\models p$ for some $p \in B(\pi)^+$ and $J \models p$ for some $p \in H(\pi)^+$, then $p \in B_{\mathsf{SE}}(\mathcal{M})^+$ and $p \notin I$ and $J \cap H_{\mathsf{SE}}(\mathcal{M})^+ \neq \emptyset$, so it follows from the definition of $B_{\mathsf{SE}}(\mathcal{M})^+$ that $(I, J) \in \mathcal{M}$.

c) If $J \not\models p$ for some $p \in B(\pi)^+$, then $p \in B_{\mathsf{SE}}(\mathcal{M})^+$ and $p \notin J$, so it follows from the definition of $B_{\mathsf{SE}}(\mathcal{M})^+$ that $(I, J) \in \mathcal{M}$.

d) If $J \models {\sim}p$ for some $p \in H(\pi)^-$, then $p \in H_{\mathsf{SE}}(\mathcal{M})^-$ and $p \notin J$, so it follows from the definition of $H_{\mathsf{SE}}(\mathcal{M})^-$ that $(I, J) \in \mathcal{M}$.

e) If $J \not\models {\sim}p$ for some $p \in B(\pi)^-$, then $p \in B_{\mathsf{SE}}(\mathcal{M})^-$ and $p \in J$, so it follows from the definition of $B_{\mathsf{SE}}(\mathcal{M})^-$ that $(I, J) \in \mathcal{M}$. □

Proposition E.14. *A set of three-valued interpretations $\mathcal{M}$ is SE-rule-expressible if and only if $\mathcal{M} \subseteq [\![\|\mathcal{M}\|_{\mathsf{SE}}]\!]_{\mathsf{SE}}$.*

Proof. If $\mathcal{M}$ is an SE-rule-expressible set of interpretations, then there exists some rule π such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{SE}}$. By Corollary 7.14, $\|\mathcal{M}\|_{\mathsf{SE}} = \|[\![\pi]\!]_{\mathsf{SE}}\|_{\mathsf{SE}} = \mathsf{can}_{\mathsf{SE}}(\pi)$, so our goal is to prove that $[\![\pi]\!]_{\mathsf{SE}} \subseteq [\![\mathsf{can}_{\mathsf{SE}}(\pi)]\!]_{\mathsf{SE}}$. This immediately follows from Theorem 7.8.

For the converse implication, suppose that $\mathcal{M} \subseteq [\![\|\mathcal{M}\|_{\mathsf{SE}}]\!]_{\mathsf{SE}}$ and put $\pi = \|\mathcal{M}\|_{\mathsf{SE}}$. By Lemma 7.17 we know that $[\![\pi]\!]_{\mathsf{SE}} \subseteq \mathcal{M}$ and together with our assumption we obtain $[\![\pi]\!]_{\mathsf{SE}} = \mathcal{M}$. Hence $\mathcal{M}$ is SE-rule-expressible. □

Proposition E.15. *A set of three-valued interpretations $\mathcal{M}$ is SE-rule-expressible if and only if there exist convex sublattices L_1, L_2 of $(\mathfrak{I}, \subseteq)$ such that*

$$\mathfrak{X} \setminus \mathcal{M} = \{\, (I, J) \in \mathfrak{X} \mid I \in L_1 \wedge J \in L_2 \,\} \cup \{\, (I, J) \in \mathfrak{X} \mid J \in L_1 \cap L_2 \,\} \enspace .$$

Proof. Suppose that $\mathcal{M}$ is an SE-rule-expressible set of interpretations. Then there exists some rule π such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{SE}}$. Let the sets of interpretations L_1, L_2 be defined as follows:

$$L_1 = \left\{\, I \in \mathfrak{I} \;\middle|\; B(\pi)^+ \subseteq I \subseteq \mathcal{A} \setminus H(\pi)^+ \,\right\} \enspace ,$$
$$L_2 = \left\{\, J \in \mathfrak{I} \;\middle|\; H(\pi)^- \cup B(\pi)^+ \subseteq J \subseteq \mathcal{A} \setminus B(\pi)^- \,\right\} \enspace .$$

It can be straightforwardly verified that these sets are convex sublattices of $(\mathfrak{I}, \subseteq)$. It remains to prove that

$$\mathfrak{X} \setminus \mathcal{M} = \{\, (I, J) \in \mathfrak{X} \mid I \in L_1 \wedge J \in L_2 \,\} \cup \{\, (I, J) \in \mathfrak{X} \mid J \in L_1 \cap L_2 \,\} \enspace . \tag{E.9}$$

According to Lemma 7.9, $(I, J) \in \mathfrak{X} \setminus \mathcal{M}$ if and only if

1. Either $B(\pi)^+ \subseteq I \subseteq \mathcal{A} \setminus H(\pi)^+$ or $J \cap H(\pi)^+ = \emptyset$ and
2. $H(\pi)^- \cup B(\pi)^+ \subseteq J \subseteq \mathcal{A} \setminus B(\pi)^-$.

Note that the first disjunct of 1. together with condition 2. occurs if and only if $I \in L_1$ and $J \in L_2$, while the second disjunct of 1. together with 2. occurs if and only if $J \in L_1 \cap L_2$. Consequently, (E.9) is satisfied.

Now suppose that L_1, L_2 are two convex sublattices of $(\mathcal{I}, \subseteq)$ such that (E.9) holds. Let $\top_1$, $\bot_1$ be the top and bottom elements of L_1 and $\top_2$, $\bot_2$ be the top and bottom elements of L_2. Furthermore, let π be a rule of the form

$$(\mathcal{A} \setminus \top_1); {\sim}\bot_2 \leftarrow \bot_1, {\sim}(\mathcal{A} \setminus \top_2).$$

We prove that $\mathcal{M} = [\![\pi]\!]_{\mathsf{SE}}$. Take some three-valued interpretation (I, J). By Lemma E.4 we know that $(I, J) \notin [\![\pi]\!]_{\mathsf{SE}}$ if and only if either $I \not\models \pi^+$ and $J \not\models \pi^-$, or $J \not\models \pi^+$ and $J \not\models \pi^-$. This holds if and only if either

$$\begin{array}{lllllllll} B(\pi)^+ \subseteq I & \text{and} & I \cap H(\pi)^+ = \emptyset & \text{and} & J \cap B(\pi)^- = \emptyset & \text{and} & H(\pi)^- \subseteq J & & \text{or} \\ B(\pi)^+ \subseteq J & \text{and} & J \cap H(\pi)^+ = \emptyset & \text{and} & J \cap B(\pi)^- = \emptyset & \text{and} & H(\pi)^- \subseteq J & . & \end{array}$$

In other words, either

$$\begin{array}{lllllllll} \bot_1 \subseteq I & \text{and} & I \cap (\mathcal{A} \setminus \top_1) = \emptyset & \text{and} & J \cap (\mathcal{A} \setminus \top_2) = \emptyset & \text{and} & \bot_2 \subseteq J & & \text{or} \\ \bot_1 \subseteq J & \text{and} & J \cap (\mathcal{A} \setminus \top_1) = \emptyset & \text{and} & J \cap (\mathcal{A} \setminus \top_2) = \emptyset & \text{and} & \bot_2 \subseteq J & . & \end{array}$$

Equivalently, either $I \in L_1$ and $J \in L_2$, or $J \in L_1 \cap L_2$. By (E.9) this is equivalent to $(I, J) \notin \mathcal{M}$. □

Theorem 7.18. *Let $\mathcal{M}$ be a set of three-valued interpretations. Then the following conditions are equivalent:*

1. *$\mathcal{M}$ is SE-rule-expressible.*
2. *$\mathcal{M} \subseteq [\![\|\mathcal{M}\|_{\mathsf{SE}}]\!]_{\mathsf{SE}}$.*
3. *There exist convex sublattices L_1, L_2 of $(\mathcal{I}, \subseteq)$ such that*

$$\mathcal{X} \setminus \mathcal{M} = \{\, (I, J) \in \mathcal{X} \mid I \in L_1 \wedge J \in L_2 \,\} \cup \{\, (I, J) \in \mathcal{X} \mid J \in L_1 \cap L_2 \,\} \quad .$$

Proof. Follows from Propositions E.14 and E.15. □

E.2 Robust Equivalence Models

We first concentrate on a systematic analysis of the expressivity of RE-models w.r.t. a single rule. The presentation follows the same pattern as the one used in Section E.1 for SE-models.

In Section E.2.1 we introduce a set of representatives of rule equivalence classes induced by RE-models and show how the representative of a class can be constructed given one of its members. Then we show how to reconstruct a representative from the set of its RE-models in Section E.2.2. Finally, in Section E.2.3, we pinpoint the conditions under which a set of three-valued interpretations is expressible by a rule under RE-models, i.e. conditions that a set $\mathcal{M}$ must satisfy so that a rule π exists such that $[\![\pi]\!]_{\mathsf{RE}} = \mathcal{M}$. All these results pave the way towards proofs of properties of RE-models reported in Section 7.2 which are reported in Section E.2.4.

E.2.1 RE-Canonical Rules

Lemma E.16. *Let (I, J) be a three-valued interpretation and π a rule. Then,*

$$(I, J) \in [\![\pi]\!]_{\mathsf{RE}} \qquad \textit{if and only if} \qquad I \models \pi^+ \vee J \models \pi^- \ .$$

Proof. First take some RE-model (I, J) of π and suppose that $J \not\models \pi^-$. We need to prove that $I \models \pi^+$. By the definition of RE-models, $I \models \pi^J$ and by Lemma E.3, $\pi^J = \pi^+$, so indeed $I \models \pi^+$.

For the converse implication, first suppose that $J \models \pi^-$. It follows from Lemma E.3 that $\pi^J = \tau$, so $I \models \pi^J$ and thus (I, J) is an RE-model of π. If $J \not\models \pi^-$, then $\pi^J = \pi^+$, so it immediately follows from $I \models \pi^+$ that (I, J) is an RE-model of π. □

We start by bringing out simple but powerful transformations that simplify a given rule while preserving its RE-models. For the rest of this section we assume that H and B are sets of literals, p is an atom and L a literal. The following result summarises the conditions under which a rule is RE-tautological:

Lemma E.17. *Rules of the following forms are RE-tautological:*

$$p; H \leftarrow p, B. \qquad\qquad H; {\sim}p \leftarrow B, {\sim}p. \qquad\qquad H \leftarrow B, p, {\sim}p.$$

Proof. First assume that a rule π is of the first form and take some three-valued interpretation (I, J). We need to show that (I, J) is an RE-model of π. According to Lemma E.16, it suffices to show that $I \models \pi^+$, which follows from the fact that p belongs to both $H(\pi)^+$ and $B(\pi)^+$.

Now suppose that π is of the second form. Given a three-valued interpretation (I, J), we see that the atom p belongs to both $H(\pi)^-$ and $B(\pi)^-$, so $J \models \pi^-$. Hence we can use Lemma E.16 to conclude that (I, J) is an RE-model of π.

Finally, suppose that π takes the third form and take some three-valued interpretation (I, J). If $J \models \pi^-$, then (I, J) is an RE-model of π by Lemma E.16. On the other hand, if $J \not\models \pi^-$, then $J \models {\sim}p$ and, consequently, $I \not\models p$ because I is a subset of J. This implies that $I \models \pi^+$, so by using Lemma E.16 we can once again conclude that (I, J) is an RE-model of π. □

Lemma E.17 shows that repeating an atom in different "components" of the rule frequently causes the rule to be RE-tautological. In particular, this happens if the same atom occurs in the positive head and positive body, or in the negative head and negative body, or in the positive and negative bodies of a rule. How about the cases when the head contains a negation of a literal from the body? The following Lemmas clarify this situation:

Lemma E.18. *Rules of the following forms are RE-equivalent:*

$$p; H \leftarrow B, {\sim}p. \qquad\qquad H \leftarrow B, {\sim}p.$$

Proof. Let the first rule be denoted by π_1, the second by π_2 and take some three-valued interpretation (I, J). We need to show that (I, J) is an RE-model of π_1 if and only if it is an RE-model of π_2. According to Lemma E.16, it suffices to prove the following:

$$I \models \pi_1^+ \vee J \models \pi_1^- \qquad \text{if and only if} \qquad I \models \pi_2^+ \vee J \models \pi_2^- \ . \tag{E.10}$$

First note that $\pi_1^- = \pi_2^-$, so

$$J \models \pi_1^- \qquad \text{if and only if} \qquad J \models \pi_2^- \ . \tag{E.11}$$

Now consider two cases:

a) If $J \models \pi_1^-$, then $J \models \pi_2^-$ due to (E.11) and we can conclude that (E.10) holds.

b) If $J \not\models \pi_1^-$, then $J \not\models \pi_2^-$ due to (E.11) and (E.10) reduces to the following goal:

$$I \models \pi_1^+ \qquad \text{if and only if} \qquad I \models \pi_2^+ \ . \tag{E.12}$$

It now suffices to observe that $J \not\models \pi_1^-$ implies $J \models \sim p$ and since I is a subset of J, we can conclude that $I \not\models p$. Since π_1^+ differs from π_2^+ only in the head atom p, it follows that (E.12) is satisfied. □

Lemma E.19. *Rules of the following forms are RE-equivalent:*

$$H; \sim p \leftarrow p, B. \qquad\qquad H \leftarrow p, B.$$

Proof. Let the first rule be denoted by π_1, the second by π_2 and take some three-valued interpretation (I, J). We need to show that (I, J) is an RE-model of π_1 if and only if it is an RE-model of π_2. According to Lemma E.16, it suffices to prove the following:

$$I \models \pi_1^+ \vee J \models \pi_1^- \qquad \text{if and only if} \qquad I \models \pi_2^+ \vee J \models \pi_2^- \ . \tag{E.13}$$

First note that $\pi_1^+ = \pi_2^+$, so

$$I \models \pi_1^+ \qquad \text{if and only if} \qquad I \models \pi_2^+ \ . \tag{E.14}$$

Now consider two cases:

a) If $I \models \pi_1^+$, then $I \models \pi_2^+$ due to (E.14) and we can conclude that (E.13) holds.

b) If $I \not\models \pi_1^+$, then $I \not\models \pi_2^+$ due to (E.14) and (E.13) reduces to the following goal:

$$J \models \pi_1^- \qquad \text{if and only if} \qquad J \models \pi_2^- \ . \tag{E.15}$$

It now suffices to observe that $I \not\models \pi_1^+$ implies $I \models p$ and since I is a subset of J, we can conclude that $J \not\models \sim p$. Since π_1^- differs from π_2^- only in the head literal $\sim p$, it follows that (E.15) is satisfied. □

So if a literal is present in the body of a rule, its negation can be removed from the head without affecting its RE-models.

Until now we have seen that a rule π that has a common atom in at least two of the sets $H(\pi)^+ \cup H(\pi)^-$, $B(\pi)^+$ and $B(\pi)^-$ is either RE-tautological, or RE-equivalent to a rule where the atom is omitted from the rule's head. So such a rule is always RE-equivalent either to the canonical tautology τ, or to a rule without such repetitions. Perhaps surprisingly, repetitions in the positive and negative head cannot be simplified away. For example, over the alphabet $\mathcal{A}_p = \{ p \}$, the rule "$p; \sim p \leftarrow .$" has two RE-models, $(\emptyset, \emptyset)$ and $(\{ p \}, \{ p \})$, so it is not RE-tautological, nor is it RE-equivalent to any of the facts "$p.$" and "$\sim p.$". Actually, it is not very difficult to see that it is not RE-equivalent to *any* other rule, even over larger alphabets. So the fact that an atom is in both $H(\pi)^+$ and $H(\pi)^-$ cannot all by itself imply that some kind of RE-models preserving rule simplification is possible.

Armed with the above results, we can introduce the notion of an RE-canonical rule. Each such rule represents a different rule equivalence class induced by the RE-models semantics. In other words, every rule is RE-equivalent to exactly one RE-canonical rule.

After the definition, we provide constructive transformations which show that this is indeed the case. Note that the definition can be derived directly from the lemmas above:

Definition E.20 (RE-Canonical Rule)**.** We say that a rule π is *RE-canonical* if either it is τ, or the sets $H(\pi)^+ \cup H(\pi)^-$, $B(\pi)^+$ and $B(\pi)^-$ are pairwise disjoint.

The following transformation provides a direct way of constructing an RE-canonical rule that is RE-equivalent to a given rule π.

Definition E.21 (Transformation into an RE-Canonical Rule)**.** Given a rule π, we define the RE-canonical rule $\mathsf{can}_{\mathsf{RE}}(\pi)$ as follows:

(i) If any of the sets $H(\pi)^+ \cap B(\pi)^+$, $H(\pi)^- \cap B(\pi)^-$ and $B(\pi)^+ \cap B(\pi)^-$ is non-empty, then $\mathsf{can}_{\mathsf{RE}}(\pi) = \tau$.

(ii) If (i) does not apply, then $\mathsf{can}_{\mathsf{RE}}(\pi)$ is the rule

$$(H(\pi)^+ \setminus B(\pi)^-); {\sim}(H(\pi)^- \setminus B(\pi)^+) \leftarrow B(\pi)^+, {\sim}B(\pi)^- .$$

Correctness of the transformation follows directly from Lemmas E.17, E.18 and E.19.

Theorem E.22. *For every rule π, $[\![\pi]\!]_{\mathsf{RE}} = [\![\mathsf{can}_{\mathsf{RE}}(\pi)]\!]_{\mathsf{RE}}$.*

Proof. This can be shown by a careful iterative application of Lemmas E.17, E.18 and E.19. First observe that if $\mathsf{can}_{\mathsf{RE}}(\pi) = \tau$, then Lemma E.17 implies that π is RE-tautological, thus indeed RE-equivalent to τ.

In the principal case we can use Lemma E.18 on all atoms from $H(\pi)^+ \cap B(\pi)^-$ and remove them one by one from $H(\pi)^+$ while preserving RE-models. A similar situation occurs with atoms from $H(\pi)^- \cap B(\pi)^+$ which can be, according to Lemma E.19, removed from $H(\pi)^-$ without affecting RE-models. After these steps are performed, we obtain the rule

$$(H(\pi)^+ \setminus B(\pi)^-); {\sim}(H(\pi)^- \setminus B(\pi)^+) \leftarrow B(\pi)^+, {\sim}B(\pi)^- .$$

This is also the result of $\mathsf{can}_{\mathsf{RE}}(\pi)$. □

What remains to be proven is that no two different RE-canonical rules are RE-equivalent. In the next section we show how every RE-canonical rule can be reconstructed from the set of its RE-models. As a consequence, no two different RE-canonical rules can have the same set of RE-models.

E.2.2 Reconstructing Rules from RE-Models

In order to reconstruct a rule π from the set of its RE-models, we need to understand how exactly each literal in the rule influences its models. The following lemma provides a useful characterisation of the set of countermodels of a rule in terms of syntax:

Lemma E.23. *Let (I, J) be a three-valued interpretation and π a rule. Then $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$ if and only if the following conditions are satisfied:*

1. *$B(\pi)^+ \subseteq I \subseteq \mathcal{A} \setminus H(\pi)^+$ and*
2. *$H(\pi)^- \subseteq J \subseteq \mathcal{A} \setminus B(\pi)^-$.*

Proof. By Lemma E.16, $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$ if and only if both $I \not\models \pi^+$ and $J \not\models \pi^-$. We will show that the former is equivalent to condition 1. and the latter to condition 2. above.

We know that $I \not\models \pi^+$ if and only if

$$\forall p \in B(\pi)^+ : I \models p \qquad \text{and} \qquad \forall p \in H(\pi)^+ : I \not\models p \enspace ,$$

which is equivalent to

$$\forall p \in B(\pi)^+ : p \in I \qquad \text{and} \qquad \forall p \in H(\pi)^+ : p \notin I \enspace .$$

In other words, $B(\pi)^+ \subseteq I$ and $I \cap H(\pi)^+ = \emptyset$, equivalently to condition 1. above.

On the other hand, $J \not\models \pi^-$ holds if and only if

$$\forall {\sim}p \in {\sim}B(\pi)^- : J \models {\sim}p \qquad \text{and} \qquad \forall {\sim}p \in {\sim}H(\pi)^- : J \not\models {\sim}p \enspace ,$$

which is equivalent to

$$\forall p \in B(\pi)^- : p \notin J \qquad \text{and} \qquad \forall p \in H(\pi)^- : p \in J \enspace .$$

In other words, $J \cap B(\pi)^- = \emptyset$ and $H(\pi)^- \subseteq J$, equivalently to condition 2. above. □

The following consequences of Lemma E.23 will be useful in further proofs.

Corollary E.24. *Let π be an RE-canonical rule different from τ, put $I = B(\pi)^+$, $J = H(\pi)^- \cup B(\pi)^+$ and $J' = \mathcal{A} \setminus B(\pi)^-$, and let p be an atom. Then the following holds:*

(1) $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$.

(2) $(I, J \cup \{ p \}) \in [\![\pi]\!]_{\mathsf{RE}}$ *if and only if* $p \in B(\pi)^-$.

(3) $(I \cup \{ p \}, J \cup \{ p \}) \in [\![\pi]\!]_{\mathsf{RE}}$ *if and only if* $p \in H(\pi)^+ \cup B(\pi)^-$.

(4) $(I, J') \notin [\![\pi]\!]_{\mathsf{RE}}$.

(5) $(I \setminus \{ p \}, J') \in [\![\pi]\!]_{\mathsf{RE}}$ *if and only if* $p \in B(\pi)^+$.

Proof. All parts of the corollary follow from Lemma E.23 and the disjointness properties satisfied by RE-canonical rules. □

If we take a closer look at the conditions in Lemma E.23, we find that the presence of an atom from $B(\pi)^-$ in J guarantees that the second condition is falsified, so (I, J) is an RE-model of π, regardless of the content of I. Somewhat similar is the situation with positive head atoms – whenever an atom from $H(\pi)^+$ is present in I, the first condition is falsified and (I, J) is an RE-model of π. More formally, given a rule π, every atom $p \in B(\pi)^-$ it holds that

$$p \in J \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \tag{E.16}$$

and for every atom $p \in H(\pi)^+$ it holds that

$$p \in I \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \enspace . \tag{E.17}$$

If we restrict ourselves to RE-canonical rules different from τ, we find that these conditions are not only necessary, but, when combined properly, also sufficient to decide what atoms belong to the negative body and positive head of π.

If not stated otherwise, we assume in the rest of this section that π is an RE-canonical rule different from τ. Keeping in mind that every atom that satisfies the condition (E.16) also satisfies the condition (E.17) (because I is a subset of J), and that $B(\pi)^-$ is by definition disjoint with $H(\pi)^+$, we arrive at the following results:

Lemma E.25. *Let p be an atom. Then $p \in B(\pi)^-$ if and only if for all $(I, J) \in \mathcal{X}$,*

$$p \in J \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \enspace .$$

Proof. Suppose that $p \in B(\pi)^-$ and take some three-valued interpretation (I, J) with $p \in J$. Then $J \not\models \sim p$, so it follows that $J \models \pi^-$. Consequently, by Lemma E.16, $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$.

To prove the converse implication, let $I = B(\pi)^+$ and $J = H(\pi)^- \cup B(\pi)^+$. It follows that $(I, J \cup \{ p \}) \in [\![\pi]\!]_{\mathsf{RE}}$, so by Corollary E.24(2) we conclude that $p \in B(\pi)^-$. □

Lemma E.26. *Let p be an atom. Then $p \in H(\pi)^+$ if and only if $p \notin B(\pi)^-$ and for all $(I, J) \in \mathcal{X}$,*

$$p \in I \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \enspace .$$

Proof. Suppose that $p \in H(\pi)^+$ and take some three-valued interpretation (I, J) with $p \in I$. Then $I \models p$, so it follows that $I \models \pi^+$. Consequently, by Lemma E.16, $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$.

To prove the converse implication, let $I = B(\pi)^+$ and $J = H(\pi)^- \cup B(\pi)^+$. It follows that $(I \cup \{ p \}, J \cup \{ p \}) \in [\![\pi]\!]_{\mathsf{RE}}$, so by Corollary E.24(3) we conclude that p belongs to $H(\pi)^+ \cup B(\pi)^-$. Moreover, by the assumption we know that $p \notin B(\pi)^-$, so $p \in H(\pi)^+$. □

As can be seen from Lemma E.23, the role of atoms from $H(\pi)^-$ and $B(\pi)^+$ is dual to that of atoms from $B(\pi)^-$ and $H(\pi)^+$. Intuitively, their absence in J and in I, respectively, implies that (I, J) is an RE-model of π. It follows from the first condition of Lemma E.23 that every $p \in B(\pi)^+$ satisfies the following condition:

$$p \notin I \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \enspace . \tag{E.18}$$

Furthermore, the second condition in Lemma E.23 implies that every $p \in H(\pi)^-$ satisfies the following condition:

$$p \notin J \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \enspace . \tag{E.19}$$

These observations lead to the following results:

Lemma E.27. *Let p be an atom. Then $p \in B(\pi)^+$ if and only if for all $(I, J) \in \mathcal{X}$,*

$$p \notin I \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \enspace .$$

Proof. Suppose that $p \in B(\pi)^+$ and take some three-valued interpretation (I, J) with $p \notin I$. Then $I \not\models p$, so it follows that $I \models \pi^+$. Consequently, by Lemma E.16, $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$.

To prove the converse implication, let $I = B(\pi)^+$ and $J' = \mathcal{A} \setminus B(\pi)^-$. It follows that $(I \setminus \{ p \}, J') \in [\![\pi]\!]_{\mathsf{RE}}$, so by Corollary E.24(5) we conclude that $p \in B(\pi)^+$. □

Lemma E.28. *Let p be an atom. Then $p \in H(\pi)^-$ if and only if $p \notin B(\pi)^+$ and for all $(I, J) \in \mathcal{X}$,*

$$p \notin J \text{ implies } (I, J) \in [\![\pi]\!]_{\mathsf{RE}} \enspace .$$

Proof. Suppose that $p \in H(\pi)^-$ and take some three-valued interpretation (I, J) with $p \notin J$. Then $J \models \sim p$, so it follows that $J \models \pi^-$. Consequently, by Lemma E.16, $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$.

To prove the converse implication, let $I = B(\pi)^+$ and $J = H(\pi)^- \cup B(\pi)^+$. Corollary E.24(1) guarantees that $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$. Furthermore, by the assumption it follows that $(I \setminus \{ p \}, J \setminus \{ p \}) \in [\![\pi]\!]_{\mathsf{RE}}$. Consequently, J must differ from $J \setminus \{ p \}$, which implies that $p \in J$. Furthermore, since $J = H(\pi)^- \cup B(\pi)^+$ and $p \notin B(\pi)^+$ by assumption, we conclude that $p \in H(\pi)^-$. □

Together, the four lemmas above are sufficient to reconstruct an RE-canonical rule from its set of RE-models. The following definition sums up these results by introducing the notion of a rule RE-induced by a set of three-valued interpretations:

Definition E.29 (Rule RE-Induced by a Set of Interpretations)**.** Let $\mathcal{M}$ be a set of three-valued interpretations. The *rule RE-induced by* $\mathcal{M}$, denoted by $\|\mathcal{M}\|_{\mathsf{RE}}$, is defined as follows: If $\mathcal{M} = \mathfrak{X}$, then $\|\mathcal{M}\|_{\mathsf{RE}} = \tau$; otherwise, $\|\mathcal{M}\|_{\mathsf{RE}}$ is of the form

$$H_{\mathsf{RE}}(\mathcal{M})^+; \sim H_{\mathsf{RE}}(\mathcal{M})^- \leftarrow B_{\mathsf{RE}}(\mathcal{M})^+, \sim B_{\mathsf{RE}}(\mathcal{M})^-.$$

where

$$\begin{aligned}
B_{\mathsf{RE}}(\mathcal{M})^- &= \{ p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \in J \Longrightarrow (I, J) \in \mathcal{M} \} \enspace , \\
H_{\mathsf{RE}}(\mathcal{M})^+ &= \{ p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \in I \Longrightarrow (I, J) \in \mathcal{M} \} \setminus B_{\mathsf{RE}}(\mathcal{M})^- \enspace , \\
B_{\mathsf{RE}}(\mathcal{M})^+ &= \{ p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \notin I \Longrightarrow (I, J) \in \mathcal{M} \} \enspace , \\
H_{\mathsf{RE}}(\mathcal{M})^- &= \{ p \in \mathcal{A} \mid \forall (I, J) \in \mathfrak{X} : p \notin J \Longrightarrow (I, J) \in \mathcal{M} \} \setminus B_{\mathsf{RE}}(\mathcal{M})^+ \enspace .
\end{aligned}$$

The main property of RE-induced rules is that every RE-canonical rule is induced by its own set of RE-models and can thus be “reconstructed” from its set of RE-models. This follows directly from Definition E.29 and Lemmas E.25 – E.28.

Theorem E.30. *For every RE-canonical rule* π, $\|[\![\pi]\!]_{\mathsf{RE}}\|_{\mathsf{RE}} = \pi$.

Proof. If $\pi = \tau$, then $[\![\pi]\!]_{\mathsf{RE}} = \mathfrak{X}$ and, by Definition E.29, $\|\mathfrak{X}\|_{\mathsf{RE}} = \tau = \pi$, so the identity is satisfied. In the principal case, π is an RE-canonical rule different from τ. Let $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. It follows from Definition E.29 and Lemmas E.25 – E.28 that $\pi = \|\mathcal{M}\|_{\mathsf{RE}}$. □

This result, together with Theorem E.22, has a number of consequences. First, for any rule π, the RE-canonical rule $\mathsf{can}_{\mathsf{RE}}(\pi)$ is RE-induced by the set of RE-models of π.

Corollary E.31. *For every rule* π, $\|[\![\pi]\!]_{\mathsf{RE}}\|_{\mathsf{RE}} = \mathsf{can}_{\mathsf{RE}}(\pi)$.

Proof. Follows directly from Theorems E.22 and E.30. □

Furthermore, Theorem E.30 implies that for two different RE-canonical rules π_1, π_2 we have $\|[\![\pi_1]\!]_{\mathsf{RE}}\|_{\mathsf{RE}} = \pi_1$ and $\|[\![\pi_2]\!]_{\mathsf{RE}}\|_{\mathsf{RE}} = \pi_2$, so $[\![\pi_1]\!]_{\mathsf{RE}}$ and $[\![\pi_2]\!]_{\mathsf{RE}}$ must differ.

Corollary E.32. *No two different RE-canonical rules are RE-equivalent.*

Proof. Follows directly from Theorem E.30. □

Finally, the previous result together with Theorem E.22 imply that for every rule there not only exists an RE-equivalent RE-canonical rule, but this rule is also unique.

Corollary E.33. *Every rule is RE-equivalent to exactly one RE-canonical rule.*

Proof. Follows directly from Theorem E.22 and Corollary E.32. □

E.2.3 RE-Rule-Expressible Sets of Interpretations

Naturally, not all sets of three-valued interpretations can be expressed by a single rule under the RE-models semantics – otherwise any program could be reduced to a single rule. We say that a set of three-valued interpretations $\mathcal{M}$ is *RE-rule-expressible* if there exists a rule π such that $[\![\pi]\!]_{\mathsf{RE}} = \mathcal{M}$. In the following we examine the conditions under which a set of three-valued interpretations is RE-rule-expressible.

We offer two approaches to find a characterisation of the class of RE-rule-expressible sets of interpretations. The first is based on RE-induced rules introduced previously, while the second is formulated using lattice theory and is strongly related to Lemma E.23.

The first characterisation follows from two properties of the $\|\cdot\|_{\mathsf{RE}}$ transformation. First, it can be applied to any set of interpretations, even those that are not RE-rule-expressible, and it always results in an RE-canonical rule. Second, if $\|\mathcal{M}\|_{\mathsf{RE}} = \pi$, then $[\![\pi]\!]_{\mathsf{RE}}$ is a subset of $\mathcal{M}$.

Lemma E.34. *Let $\mathcal{M}$ be a set of three-valued interpretations different from $\mathfrak{X}$. Then the sets $H_{\mathsf{RE}}(\mathcal{M})^+ \cup H_{\mathsf{RE}}(\mathcal{M})^-$, $B_{\mathsf{RE}}(\mathcal{M})^+$ and $B_{\mathsf{RE}}(\mathcal{M})^-$ are pairwise disjoint.*

Proof. Suppose first that p is a member of both $H_{\mathsf{RE}}(\mathcal{M})^+ \cup H_{\mathsf{RE}}(\mathcal{M})^-$ and $B_{\mathsf{RE}}(\mathcal{M})^+$. Since p belongs to $B_{\mathsf{RE}}(\mathcal{M})^+$, it cannot belong to $H_{\mathsf{RE}}(\mathcal{M})^-$ by definition. So p belongs to both $H_{\mathsf{RE}}(\mathcal{M})^+$ and $B_{\mathsf{RE}}(\mathcal{M})^+$. Take an arbitrary three-valued interpretation (I, J). If $p \in I$, then, by the definition of $H_{\mathsf{RE}}(\mathcal{M})^+$, $(I, J) \in \mathcal{M}$. If $p \notin I$, then, by the definition of $B_{\mathsf{RE}}(\mathcal{M})^+$, $(I, J) \in \mathcal{M}$. As a consequence, $\mathcal{M} = \mathfrak{X}$, contrary to our assumption.

Now suppose that p is a member of both $H_{\mathsf{RE}}(\mathcal{M})^+ \cup H_{\mathsf{RE}}(\mathcal{M})^-$ and $B_{\mathsf{RE}}(\mathcal{M})^-$. Then, since p belongs to $B_{\mathsf{RE}}(\mathcal{M})^-$, it cannot belong to $H_{\mathsf{RE}}(\mathcal{M})^+$ by definition. So p belongs to both $H_{\mathsf{RE}}(\mathcal{M})^-$ and $B_{\mathsf{RE}}(\mathcal{M})^-$. Take an arbitrary three-valued interpretation (I, J). If $p \in J$, then, by the definition of $B_{\mathsf{RE}}(\mathcal{M})^-$, $(I, J) \in \mathcal{M}$. If $p \notin J$, then, by the definition of $H_{\mathsf{RE}}(\mathcal{M})^-$, $(I, J) \in \mathcal{M}$. As a consequence, $\mathcal{M} = \mathfrak{X}$, contrary to our assumption.

Finally, suppose that p is a member of both $B_{\mathsf{RE}}(\mathcal{M})^+$ and $B_{\mathsf{RE}}(\mathcal{M})^-$ and take an arbitrary three-valued interpretation (I, J). If $p \in J$, then, by the definition of $B_{\mathsf{RE}}(\mathcal{M})^-$, $(I, J) \in \mathcal{M}$. If $p \notin J$, then $p \notin I$ because I is a subset of J, and by the definition of $B_{\mathsf{RE}}(\mathcal{M})^+$ we obtain $(I, J) \in \mathcal{M}$ once again. Consequently, $\mathcal{M} = \mathfrak{X}$, contrary to our assumption. □

Lemma E.35. *For every set of three-valued interpretations $\mathcal{M}$, $\|\mathcal{M}\|_{\mathsf{RE}}$ is an RE-canonical rule.*

Proof. If $\mathcal{M} = \mathfrak{X}$, then $\|\mathcal{M}\|_{\mathsf{RE}} = \tau$ and the proof is finished. Otherwise, $\|\mathcal{M}\|_{\mathsf{RE}}$ is of the form

$$H_{\mathsf{RE}}(\mathcal{M})^+; {\sim}H_{\mathsf{RE}}(\mathcal{M})^- \leftarrow B_{\mathsf{RE}}(\mathcal{M})^+, {\sim}B_{\mathsf{RE}}(\mathcal{M})^-.$$

To show that this rule is RE-canonical, we need to prove that the sets $H_{\mathsf{RE}}(\mathcal{M})^+ \cup H_{\mathsf{RE}}(\mathcal{M})^-$, $B_{\mathsf{RE}}(\mathcal{M})^+$ and $B_{\mathsf{RE}}(\mathcal{M})^-$ are pairwise disjoint, which follows from Lemma E.34. □

Lemma E.36. *The set of all RE-models of an RE-canonical rule π is the least among all sets of three-valued interpretations $\mathcal{M}$ such that $\|\mathcal{M}\|_{\mathsf{RE}} = \pi$.*

Proof. Let π be an RE-canonical rule. From Theorem E.30 we know that $\|[\![\pi]\!]_{\mathsf{RE}}\|_{\mathsf{RE}} = \pi$, so it remains to show that $[\![\pi]\!]_{\mathsf{RE}}$ is a subset of every set of interpretations $\mathcal{M}$ such that $\|\mathcal{M}\|_{\mathsf{RE}} = \pi$. Take one such $\mathcal{M}$. If $\pi = \tau$, then $[\![\pi]\!]_{\mathsf{RE}} = \mathfrak{X}$. Furthermore, $\mathcal{M} = \mathfrak{X}$, for otherwise the set $H_{\mathsf{RE}}(\mathcal{M})^+ \cap B_{\mathsf{RE}}(\mathcal{M})^+$ would be non-empty, contrary to Lemma E.34.

In the principal case, π is not the canonical tautology, so from $\|\mathcal{M}\|_{\mathsf{RE}} = \pi$ we obtain that $\mathcal{M} \neq \mathfrak{X}$ as well as $H(\pi)^+ = H_{\mathsf{RE}}(\mathcal{M})^+$, $H(\pi)^- = H_{\mathsf{RE}}(\mathcal{M})^-$, $B(\pi)^+ = B_{\mathsf{RE}}(\mathcal{M})^+$ and $B(\pi)^- = B_{\mathsf{RE}}(\mathcal{M})^-$. Take some $(I, J) \in [\![\pi]\!]_{\mathsf{RE}}$, we need to show that $(I, J) \in \mathcal{M}$. By

Lemma E.16 we obtain that either $I \models \pi^+$ or $J \models \pi^-$. This means that we have four cases to consider:

a) If $I \not\models p$ for some $p \in B(\pi)^+$, then $p \in B_{\mathsf{RE}}(\mathcal{M})^+$ and $p \notin I$, so it follows from the definition of $B_{\mathsf{RE}}(\mathcal{M})^+$ that $(I, J) \in \mathcal{M}$.

b) If $I \models p$ for some $p \in H(\pi)^+$, then $p \in H_{\mathsf{RE}}(\mathcal{M})^+$ and $p \in I$, so it follows from the definition of $H_{\mathsf{RE}}(\mathcal{M})^+$ that $(I, J) \in \mathcal{M}$.

c) If $J \not\models \sim p$ for some $p \in B(\pi)^-$, then $p \in B_{\mathsf{RE}}(\mathcal{M})^-$ and $p \in J$, so it follows from the definition of $B_{\mathsf{RE}}(\mathcal{M})^-$ that $(I, J) \in \mathcal{M}$.

d) If $J \models \sim p$ for some $p \in H(\pi)^-$, then $p \in H_{\mathsf{RE}}(\mathcal{M})^-$ and $p \notin J$, so it follows from the definition of $H_{\mathsf{RE}}(\mathcal{M})^-$ that $(I, J) \in \mathcal{M}$. □

Thus, to verify that $\mathcal{M}$ is RE-rule-expressible, it suffices to check that all interpretations from $\mathcal{M}$ are RE-models of $\|\mathcal{M}\|_{\mathsf{RE}}$.

Proposition E.37. *A set of three-valued interpretations $\mathcal{M}$ is RE-rule-expressible if and only if $\mathcal{M} \subseteq [\![\|\mathcal{M}\|_{\mathsf{RE}}]\!]_{\mathsf{RE}}$.*

Proof. If $\mathcal{M}$ is an RE-rule-expressible set of interpretations, then there exists some rule π such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. By Corollary E.31, $\|\mathcal{M}\|_{\mathsf{RE}} = \|[\![\pi]\!]_{\mathsf{RE}}\|_{\mathsf{RE}} = \mathsf{can}_{\mathsf{RE}}(\pi)$, so our goal is to prove that $[\![\pi]\!]_{\mathsf{RE}} \subseteq [\![\mathsf{can}_{\mathsf{RE}}(\pi)]\!]_{\mathsf{RE}}$. This immediately follows from Theorem E.22.

For the converse implication, suppose that $\mathcal{M} \subseteq [\![\|\mathcal{M}\|_{\mathsf{RE}}]\!]_{\mathsf{RE}}$ and put $\pi = \|\mathcal{M}\|_{\mathsf{RE}}$. By Lemma E.36 we know that $[\![\pi]\!]_{\mathsf{RE}} \subseteq \mathcal{M}$ and together with our assumption we obtain $[\![\pi]\!]_{\mathsf{RE}} = \mathcal{M}$. Hence $\mathcal{M}$ is RE-rule-expressible. □

The second characterisation follows from Lemma E.23 which tells us that if $\mathcal{M}$ is RE-rule-expressible, then its complement consists of interpretations (I, J) following a certain pattern. Their second component J always includes a fixed set of atoms and is itself included in another fixed set of atoms. Their first component I satisfies a similar property. More formally, for the sets

$$I^\perp = B(\pi)^+, \qquad I^\top = \mathcal{A} \setminus H(\pi)^+,$$
$$J^\perp = H(\pi)^- \cup B(\pi)^+, \qquad J^\top = \mathcal{A} \setminus B(\pi)^-,$$

it holds that all interpretations from $\mathcal{X} \setminus \mathcal{M}$ are of the form (I, J) where $J^\perp \subseteq J \subseteq J^\top$ and $I^\perp \subseteq I \subseteq I^\top$. It turns out that this also holds vice versa: if the $\mathcal{X} \setminus \mathcal{M}$ satisfies the above property, then $\mathcal{M}$ is RE-rule-expressible. Furthermore, to accentuate the particular structure that arises, we can substitute the condition $J^\perp \subseteq J \subseteq J^\top$ with saying that J belongs to a convex sublattice of $\mathcal{I}$.[1] A similar substitution can be performed for I, yielding:

Proposition E.38. *A set of three-valued interpretations $\mathcal{M}$ is RE-rule-expressible if and only if there exist convex sublattices L_1, L_2 of $(\mathcal{I}, \subseteq)$ such that*

$$\mathcal{X} \setminus \mathcal{M} = \{ (I, J) \in \mathcal{X} \mid I \in L_1 \wedge J \in L_2 \} \enspace .$$

[1] A sublattice L of L' is *convex* if $u \in L$ whenever $s, t \in L$ and $s \leq u \leq t$ holds in L'. For more details see e.g. (Davey and Priestley, 1990).

Proof. Suppose that $\mathcal{M}$ is an RE-rule-expressible set of interpretations. Then there exists some rule π such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. Let the sets of interpretations L_1, L_2 be defined as follows:

$$L_1 = \{ I \in \mathcal{I} \mid B(\pi)^+ \subseteq I \subseteq \mathcal{A} \setminus H(\pi)^+ \}$$
$$L_2 = \{ J \in \mathcal{I} \mid H(\pi)^- \subseteq J \subseteq \mathcal{A} \setminus B(\pi)^- \}$$

It can be straightforwardly verified that these sets are convex sublattices of $(\mathcal{I}, \subseteq)$. It remains to prove that

$$\mathcal{X} \setminus \mathcal{M} = \{ (I, J) \in \mathcal{X} \mid I \in L_1 \land J \in L_2 \} \quad . \tag{E.20}$$

This follows directly by Lemma E.23.

Now suppose that L_1, L_2 are two convex sublattices of $(\mathcal{I}, \subseteq)$ such that (E.20) holds. Let $\top_1$, $\bot_1$ be the top and bottom elements of L_1 and $\top_2$, $\bot_2$ be the top and bottom elements of L_2. Furthermore, let π be a rule of the form

$$(\mathcal{A} \setminus \top_1); \sim\bot_2 \leftarrow \bot_1, \sim(\mathcal{A} \setminus \top_2).$$

We prove that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. Take some three-valued interpretation (I, J). By Lemma E.16 we know that $(I, J) \notin [\![\pi]\!]_{\mathsf{RE}}$ if and only if $I \not\models \pi^+$ and $J \not\models \pi^-$ which holds if and only if $B(\pi)^+ \subseteq I$ and $I \cap H(\pi)^+ = \emptyset$ and $J \cap B(\pi)^- = \emptyset$ and $H(\pi)^- \subseteq J$. In other words,

$$\bot_1 \subseteq I \quad \text{and} \quad I \cap (\mathcal{A} \setminus \top_1) = \emptyset \quad \text{and} \quad J \cap (\mathcal{A} \setminus \top_2) = \emptyset \quad \text{and} \quad \bot_2 \subseteq J \ .$$

Equivalently, $I \in L_1$ and $J \in L_2$, so by (E.20) it is also equivalent to $(I, J) \notin \mathcal{M}$. □

Theorem E.39. *Let $\mathcal{M}$ be a set of three-valued interpretations. Then the following conditions are equivalent:*

1. *$\mathcal{M}$ is RE-rule-expressible.*
2. *$\mathcal{M} \subseteq [\![\,\|\mathcal{M}\|_{\mathsf{RE}}\,]\!]_{\mathsf{RE}}$.*
3. *There exist convex sublattices L_1, L_2 of $(\mathcal{I}, \subseteq)$ such that*

$$\mathcal{X} \setminus \mathcal{M} = \{ (I, J) \in \mathcal{X} \mid I \in L_1 \land J \in L_2 \} \quad .$$

Proof. Follows from Propositions E.37 and E.38. □

E.2.4 Comparison of RE-Models with SE-Models

Proposition 7.21. *If π, σ are two different abolishing rules or an abolishing rule and a constraint, then π, σ are not RE-equivalent.*

Proof. Follows from Corollary E.32 since every abolishing rule and every constraint is RE-canonical. □

Lemma E.40. *For every rule π, $\mathsf{can}_{\mathsf{SE}}(\pi) = \mathsf{can}_{\mathsf{SE}}(\mathsf{can}_{\mathsf{RE}}(\pi))$.*

Proof. Follows directly from Definitions 7.7 and E.21. □

Lemma E.41. *If π is not RE-equivalent to any abolishing rule, then $\mathsf{can}_{\mathsf{RE}}(\pi) = \mathsf{can}_{\mathsf{SE}}(\pi)$.*

Proof. Follows directly from Definitions 7.7 and E.21. □

Proposition 7.22.

- *If two rules are* RE*-equivalent, then they are* SE*-equivalent.*
- *If two rules, neither of which is* RE*-equivalent to an abolishing rule, are* SE*-equivalent, then they are* RE*-equivalent.*
- *A rule is* RE*-tautological if and only if it is* SE*-tautological.*

Proof. Suppose that π and σ are RE-equivalent. Then $\mathsf{can}_{\mathsf{RE}}(\pi) = \mathsf{can}_{\mathsf{RE}}(\sigma)$ by Theorem E.22 and Corollary E.32. By Lemma E.40 it follows that

$$\mathsf{can}_{\mathsf{SE}}(\pi) = \mathsf{can}_{\mathsf{SE}}(\mathsf{can}_{\mathsf{RE}}(\pi)) = \mathsf{can}_{\mathsf{SE}}(\mathsf{can}_{\mathsf{RE}}(\sigma)) = \mathsf{can}_{\mathsf{SE}}(\sigma)$$

and by Theorem 7.8 we can conclude that π, σ are SE-equivalent.

Now suppose that neither π nor σ is RE-equivalent to an abolishing rule and π is RE-equivalent to σ. Then, by Theorem E.22, $\mathsf{can}_{\mathsf{RE}}(\pi)$ is RE-equivalent to $\mathsf{can}_{\mathsf{RE}}(\sigma)$ and by Corollary E.32, $\mathsf{can}_{\mathsf{RE}}(\pi) = \mathsf{can}_{\mathsf{RE}}(\sigma)$. Furthermore, it follows from Lemma E.41 that $\mathsf{can}_{\mathsf{SE}}(\pi) = \mathsf{can}_{\mathsf{SE}}(\sigma)$ and by Theorem 7.8, π is SE-equivalent to σ.

Finally, by Corollary E.32, a rule π is RE-tautological if and only if $\mathsf{can}_{\mathsf{RE}}(\pi) = \tau$ which holds if and only if $\mathsf{can}_{\mathsf{SE}}(\pi) = \tau$ (c.f. Definitions 7.7 and E.21) which, by Corollary 7.15, holds if and only if π is SE-tautological. □

Proposition 7.23. *An interpretation J is a stable model of a program $\mathcal{P}$ if and only if $(J, J) \in [\![\mathcal{P}]\!]_{\mathsf{RE}}$ and for all $I \subsetneq J$, $(I, J) \notin [\![\mathcal{P}]\!]_{\mathsf{RE}}$.*

Proof. Suppose that J is a stable model of $\mathcal{P}$. Then J is a subset-minimal model of $\mathcal{P}^J$. Thus, (J, J) is an RE-model of $\mathcal{P}$. Now suppose that (I, J) is an RE-model of $\mathcal{P}$ for some $I \subseteq J$. Then $I \models \mathcal{P}^J$ and by the minimality of J we then obtain that $I = J$.

Now suppose that (J, J) is an RE-model of $\mathcal{P}$ and for all $I \subsetneq J$, (I, J) is not an RE-model of $\mathcal{P}$. Then $J \models \mathcal{P}^J$. Furthermore, it must also be a subset-minimal model of $\mathcal{P}^J$. Consequently, J is a stable model of $\mathcal{P}$. □

Proposition 7.24. *Let $\mathcal{M}$ be a set of three-valued interpretations. Then there exists a program $\mathcal{P}$ such that $[\![\mathcal{P}]\!]_{\mathsf{RE}} = \mathcal{M}$.*

Proof. Let $\mathcal{P}$ contain the rule

$$\pi_{(I,J)} : \quad (\mathcal{A} \setminus I); {\sim}J \leftarrow I, {\sim}(\mathcal{A} \setminus J).$$

for all three-valued interpretations (I, J) that do not belong to $\mathcal{M}$. It is an immediate consequence of Lemma E.23 that $[\![\pi_{(I,J)}]\!]_{\mathsf{RE}} = \mathcal{X} \setminus \{ (I, J) \}$. Thus,

$$[\![\mathcal{P}]\!]_{\mathsf{RE}} = \bigcap_{(I,J) \in \mathcal{X} \setminus \mathcal{M}} \mathcal{X} \setminus \{ (I, J) \} = \mathcal{X} \setminus \bigcup_{(I,J) \in \mathcal{X} \setminus \mathcal{M}} \{ (I, J) \} = \mathcal{X} \setminus (\mathcal{X} \setminus \mathcal{M}) = \mathcal{M} \ . \qquad \square$$

E.3 Program Equivalence and Entailment

Definition E.42. A *program entailment relation* is a preorder on the set of all programs. A *program equivalence relation* is an equivalence relation on the set of all programs.

Given a program entailment relation $\models$ and a program equivalence relation $\equiv$, we say that $\models$ *is associated with* $\equiv$ if for all programs $\mathcal{P}$, $\mathcal{Q}$,

$$\mathcal{P} \equiv \mathcal{Q} \qquad \text{if and only if} \qquad \mathcal{P} \models \mathcal{Q} \text{ and } \mathcal{Q} \models \mathcal{P} \ .$$

Proposition 7.28. *If X is one of SE, RE, SMR, RMR, SR, RR and SU, then the program entailment relation* $\models_{\mathsf{X}}$ *is associated with the program equivalence relation* $\equiv_{\mathsf{X}}$.

Proof. If X is SE, RE or SU, then the property follows immediately from the definitions of $\models_{\mathsf{X}}$ and $\equiv_{\mathsf{X}}$.

If X is either SR or RR, then it follows from the definition of $\models_{\mathsf{X}}$ that $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ is equivalent to $\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{X}} \supseteq \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{X}}$. Thus, $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ together with $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$ is equivalent to $\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{X}} = \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{X}}$, which is the definition of $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$.

It remains to consider the case when X is SMR or RMR. Let Y be SE or RE, respectively. First suppose that $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$. By the definition of $\equiv_{\mathsf{X}}$ we obtain that

$$\min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} = \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}} \quad . \tag{E.21}$$

Our goal is to prove that $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ and $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$. We only show the former; the proof of the latter is analogous. Take some $\sigma \in \mathcal{Q}$. Our goal is find some $\pi \in \mathcal{P}^\tau$ such that $[\![\pi]\!]_{\mathsf{Y}} \subseteq [\![\sigma]\!]_{\mathsf{Y}}$. Take some subset-minimal set of three-valued interpretations $\mathcal{M} \in \langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$ such that $\mathcal{M} \subseteq [\![\sigma]\!]_{\mathsf{Y}}$. It follows from (E.21) that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}}$. In other words, there exists some $\pi \in \mathcal{P}^\tau$ such that $[\![\pi]\!]_{\mathsf{Y}} = \mathcal{M} \subseteq [\![\sigma]\!]_{\mathsf{Y}}$.

Now suppose that both $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ and $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$. We need to prove that $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$, i.e. that $\min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} = \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$. We only show that $\min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} \subseteq \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$; the proof of the other inclusion is analogical. Take some $\pi \in \mathcal{P}^\tau$ such that

$$[\![\pi]\!]_{\mathsf{Y}} \in \min\langle\!\langle \mathcal{P}^\tau \rangle\!\rangle_{\mathsf{Y}} \quad . \tag{E.22}$$

Since $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$, there exists some $\sigma \in \mathcal{Q}^\tau$ such that

$$[\![\sigma]\!]_{\mathsf{Y}} \subseteq [\![\pi]\!]_{\mathsf{Y}} \quad . \tag{E.23}$$

Let $\sigma' \in \mathcal{Q}^\tau$ be such that

$$[\![\sigma']\!]_{\mathsf{Y}} \in \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}} \qquad \text{and} \qquad [\![\sigma']\!]_{\mathsf{Y}} \subseteq [\![\sigma]\!]_{\mathsf{Y}} \quad . \tag{E.24}$$

Since $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$, there exists some $\pi' \in \mathcal{P}^\tau$ such that

$$[\![\pi']\!]_{\mathsf{Y}} \subseteq [\![\sigma']\!]_{\mathsf{Y}} \quad . \tag{E.25}$$

By (E.25), (E.24) and (E.23) we now obtain

$$[\![\pi']\!]_{\mathsf{Y}} \subseteq [\![\sigma']\!]_{\mathsf{Y}} \subseteq [\![\sigma]\!]_{\mathsf{Y}} \subseteq [\![\pi]\!]_{\mathsf{Y}} \quad ,$$

so by (E.22) we can conclude that

$$[\![\pi']\!]_{\mathsf{Y}} = [\![\sigma']\!]_{\mathsf{Y}} = [\![\sigma]\!]_{\mathsf{Y}} = [\![\pi]\!]_{\mathsf{Y}} \quad ,$$

Consequently, it follows from (E.24) that $[\![\pi]\!]_{\mathsf{Y}} \in \min\langle\!\langle \mathcal{Q}^\tau \rangle\!\rangle_{\mathsf{Y}}$. □

Lemma E.43. *Let* $\models_{\mathsf{X}}$, $\models_{\mathsf{Y}}$ *be program entailment relations and* $\equiv_{\mathsf{X}}$, $\equiv_{\mathsf{Y}}$ *program equivalence relations such that* $\models_{\mathsf{X}}$ *is associated with* $\equiv_{\mathsf{X}}$ *and* $\models_{\mathsf{Y}}$ *is associated with* $\equiv_{\mathsf{Y}}$. *The following holds:*

$$\models_{\mathsf{X}} \preceq \models_{\mathsf{Y}} \qquad \textit{implies} \qquad \equiv_{\mathsf{X}} \preceq \equiv_{\mathsf{Y}} \quad .$$

Proof. Suppose that $\models_{\mathsf{X}} \preceq \models_{\mathsf{Y}}$ and take some programs $\mathcal{P}$, $\mathcal{Q}$ such that $\mathcal{P} \equiv_{\mathsf{Y}} \mathcal{Q}$. We need to show that $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$. Since $\models_{\mathsf{Y}}$ is associated with $\equiv_{\mathsf{Y}}$, we can conclude that $\mathcal{P} \models_{\mathsf{Y}} \mathcal{Q}$

and $\mathcal{Q} \models_{\mathsf{Y}} \mathcal{P}$. Furthermore, from $\models_{\mathsf{X}} \preceq \models_{\mathsf{Y}}$ it follows that $\mathcal{P} \models_{\mathsf{X}} \mathcal{Q}$ and $\mathcal{Q} \models_{\mathsf{X}} \mathcal{P}$, and the assumption that $\models_{\mathsf{X}}$ is associated with $\equiv_{\mathsf{X}}$ implies $\mathcal{P} \equiv_{\mathsf{X}} \mathcal{Q}$. □

Corollary E.44. *Let $\models_{\mathsf{X}}$, $\models_{\mathsf{Y}}$ be program entailment relations and $\equiv_{\mathsf{X}}$, $\equiv_{\mathsf{Y}}$ program equivalence relations such that $\models_{\mathsf{X}}$ is associated with $\equiv_{\mathsf{X}}$ and $\models_{\mathsf{Y}}$ is associated with $\equiv_{\mathsf{Y}}$. The following holds:*

$$\equiv_{\mathsf{X}} \prec \equiv_{\mathsf{Y}} \textit{ and } \models_{\mathsf{X}} \prec \models_{\mathsf{Y}} \qquad \textit{if and only if} \qquad \models_{\mathsf{X}} \preceq \models_{\mathsf{Y}} \textit{ and } \equiv_{\mathsf{Y}} \not\preceq \equiv_{\mathsf{X}} \quad .$$

Proof. By the definition, $\equiv_{\mathsf{X}} \prec \equiv_{\mathsf{Y}}$ and $\models_{\mathsf{X}} \prec \models_{\mathsf{Y}}$ hold if and only if

$$\equiv_{\mathsf{X}} \preceq \equiv_{\mathsf{Y}} \text{ and } \equiv_{\mathsf{Y}} \not\preceq \equiv_{\mathsf{X}} \text{ and } \models_{\mathsf{X}} \preceq \models_{\mathsf{Y}} \text{ and } \models_{\mathsf{Y}} \not\preceq \models_{\mathsf{X}} \quad . \tag{E.26}$$

By Lemma E.43, $\models_{\mathsf{X}} \preceq \models_{\mathsf{Y}}$ implies $\equiv_{\mathsf{X}} \preceq \equiv_{\mathsf{Y}}$ and $\equiv_{\mathsf{Y}} \not\preceq \equiv_{\mathsf{X}}$ implies $\models_{\mathsf{Y}} \not\preceq \equiv_{\mathsf{X}}$, so condition (E.26) can be simplified to

$$\models_{\mathsf{X}} \preceq \models_{\mathsf{Y}} \text{ and } \equiv_{\mathsf{Y}} \not\preceq \equiv_{\mathsf{X}} \quad .$$ □

Lemma E.45. *Let π be a rule and J an interpretation. The following holds:*

$$J \models \pi \qquad \textit{if and only if} \qquad J \models \pi^J \qquad \textit{if and only if} \qquad (J, J) \in [\![\pi]\!]_{\mathsf{RE}} \quad .$$

Proof. Follows from Lemmas E.2 and E.3 and from the definition of RE-models. □

Lemma E.46. *Let Π, Σ be rules or programs. The following holds:*

$$[\![\Pi]\!]_{\mathsf{RE}} \subseteq [\![\Sigma]\!]_{\mathsf{RE}} \qquad \textit{implies} \qquad [\![\Pi]\!]_{\mathsf{SE}} \subseteq [\![\Sigma]\!]_{\mathsf{SE}} \quad .$$

Proof. Assume that $[\![\Pi]\!]_{\mathsf{RE}} \subseteq [\![\Sigma]\!]_{\mathsf{RE}}$. Then for all three-valued interpretations (I, J),

$$I \models \Pi^J \qquad \text{implies} \qquad I \models \Sigma^J \quad . \tag{E.27}$$

Together with Lemma E.45 this implies that for all interpretations J,

$$J \models \Pi \qquad \text{implies} \qquad J \models \Pi^J \qquad \text{implies} \qquad J \models \Sigma^J \qquad \text{implies} \qquad J \models \Sigma \quad . \tag{E.28}$$

In order to show that $[\![\Pi]\!]_{\mathsf{SE}} \subseteq [\![\Sigma]\!]_{\mathsf{SE}}$, take some $(I, J) \in [\![\Pi]\!]_{\mathsf{SE}}$. By the definition of SE-models,

$$J \models \Pi \qquad \text{and} \qquad I \models \Pi^J \quad ,$$

so by (E.28) and (E.27) we can conclude that

$$J \models \Sigma \qquad \text{and} \qquad I \models \Sigma^J \quad .$$

Thus, by the definition of SE-models, $(I, J) \in [\![\Sigma]\!]_{\mathsf{SE}}$. □

Proposition 7.30. *The following holds:*

(1) $\equiv_{\mathsf{SM}} \prec \equiv_{\mathsf{SE}} \prec \equiv_{\mathsf{RE}} \prec \equiv_{\mathsf{RMR}} \prec \equiv_{\mathsf{RR}} \prec \equiv_{\mathsf{SU}}$ *and* $\models_{\mathsf{SE}} \prec \models_{\mathsf{RE}} \prec \models_{\mathsf{RMR}} \prec \models_{\mathsf{RR}} \prec \models_{\mathsf{SU}}$;

(2) $\equiv_{\mathsf{SE}} \prec \equiv_{\mathsf{SMR}} \prec \equiv_{\mathsf{SR}} \prec \equiv_{\mathsf{RR}}$ *and* $\models_{\mathsf{SE}} \prec \models_{\mathsf{SMR}} \prec \models_{\mathsf{SR}} \prec \models_{\mathsf{RR}}$;

(3) $\equiv_{\mathsf{SMR}} \prec \equiv_{\mathsf{RMR}}$ *and* $\models_{\mathsf{SMR}} \prec \models_{\mathsf{RMR}}$;

(4) $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SMR}}$ *and* $\equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{RE}}$ *and* $\models_{\mathsf{RE}} \not\preceq \models_{\mathsf{SMR}}$ *and* $\models_{\mathsf{SMR}} \not\preceq \models_{\mathsf{RE}}$;

(5) $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SR}}$ *and* $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RE}}$ *and* $\models_{\mathsf{RE}} \not\preceq \models_{\mathsf{SR}}$ *and* $\models_{\mathsf{SR}} \not\preceq \models_{\mathsf{RE}}$;

(6) $\equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{SR}}$ *and* $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RMR}}$ *and* $\models_{\mathsf{RMR}} \not\preceq \models_{\mathsf{SR}}$ *and* $\models_{\mathsf{SR}} \not\preceq \models_{\mathsf{RMR}}$.

Proof. We consider each statement separately:

(1) First we show that $\equiv_{\mathsf{SM}} \prec \equiv_{\mathsf{SE}}$, i.e. that $\equiv_{\mathsf{SM}} \preceq \equiv_{\mathsf{SE}}$ and $\equiv_{\mathsf{SE}} \not\preceq \equiv_{\mathsf{SM}}$. To verify the former, suppose that $\mathcal{P}$, $\mathcal{Q}$ are programs with $\mathcal{P} \equiv_{\mathsf{SE}} \mathcal{Q}$. Then, according to Proposition 6.4, $\mathcal{P} \cup \emptyset$ has the same stable models as $\mathcal{Q} \cup \emptyset$. Consequently, $\mathcal{P} \equiv_{\mathsf{SM}} \mathcal{Q}$. To see that $\equiv_{\mathsf{SE}} \not\preceq \equiv_{\mathsf{SM}}$, observe that the programs $\mathcal{P} = \emptyset$, $\mathcal{Q} = \{ p \leftarrow q. \}$ are SM-equivalent but not SE-equivalent.

Turning to the remaining relationships, it follows from Corollary E.44 that we can instead prove that

$$\models_{\mathsf{SE}} \preceq \models_{\mathsf{RE}} \preceq \models_{\mathsf{RMR}} \preceq \models_{\mathsf{RR}} \preceq \models_{\mathsf{SU}} \qquad \text{and} \qquad \equiv_{\mathsf{SU}} \not\preceq \equiv_{\mathsf{RR}} \not\preceq \equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SE}} \quad . \tag{E.29}$$

We first concentrate on the left-hand side of (E.29). In order to show that $\models_{\mathsf{SE}} \preceq \models_{\mathsf{RE}}$, suppose that $\mathcal{P}$, $\mathcal{Q}$ are programs such that $\mathcal{P} \models_{\mathsf{RE}} \mathcal{Q}$. Then $[\![\mathcal{P}]\!]_{\mathsf{RE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{RE}}$ and it follows from Lemma E.46 that $[\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{SE}}$. Consequently, $\mathcal{P} \models_{\mathsf{SE}} \mathcal{Q}$.

We also need to prove that $\models_{\mathsf{RE}} \preceq \models_{\mathsf{RMR}}$. Take some programs $\mathcal{P}$, $\mathcal{Q}$ with $\mathcal{P} \models_{\mathsf{RMR}} \mathcal{Q}$ and put $\mathcal{P}^{\tau} = \mathcal{P} \cup \{ \tau \}$. It follows that

$$\forall \sigma \in \mathcal{Q} \, \exists \pi_\sigma \in \mathcal{P}^{\tau} : [\![\pi_\sigma]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}} \quad . \tag{E.30}$$

We need to prove that $[\![\mathcal{P}]\!]_{\mathsf{RE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{RE}}$. Suppose that $X \in [\![\mathcal{P}]\!]_{\mathsf{RE}}$. Then for all $\pi \in \mathcal{P}$,

$$X \in [\![\pi]\!]_{\mathsf{RE}} \quad . \tag{E.31}$$

Take some $\sigma \in \mathcal{Q}$. Our goal is to show that $X \in [\![\sigma]\!]_{\mathsf{RE}}$. By (E.30) there exists some $\pi_\sigma \in \mathcal{P}^{\tau}$ such that $[\![\pi_\sigma]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}}$. If $\pi_\sigma = \tau$, then it immediately follows that $X \in \mathcal{X} = [\![\tau]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}}$. If $\pi_\sigma \in \mathcal{P}$, then $X \in [\![\pi_\sigma]\!]_{\mathsf{RE}}$ by (E.31), so that $X \in [\![\sigma]\!]_{\mathsf{RE}}$.

Our next goal is to show that $\models_{\mathsf{RMR}} \preceq \models_{\mathsf{RR}}$. This follows directly by the definitions of $\models_{\mathsf{RMR}}$ and $\models_{\mathsf{RR}}$.

To prove the final part of the left-hand side of (E.29), suppose that $\mathcal{P} \models_{\mathsf{SU}} \mathcal{Q}$. Then

$$[\![\mathcal{Q} \setminus \mathcal{P}]\!]_{\mathsf{SE}} = \mathcal{X} \quad . \tag{E.32}$$

We need to prove that $\mathcal{P} \models_{\mathsf{RR}} \mathcal{Q}$, i.e. that for every $\sigma \in \mathcal{Q}$ there is some $\pi \in \mathcal{P}^{\tau}$ such that $[\![\pi]\!]_{\mathsf{RE}} = [\![\sigma]\!]_{\mathsf{RE}}$. Pick some $\sigma \in \mathcal{Q}$. Note that $\mathcal{Q} = (\mathcal{Q} \cap \mathcal{P}) \cup (\mathcal{Q} \setminus \mathcal{P})$. If $\sigma \in \mathcal{Q} \cap \mathcal{P}$, then $\sigma \in \mathcal{P}$ and we can put $\pi = \sigma$ to finish the proof. In the remaining case, $\sigma \in \mathcal{Q} \setminus \mathcal{P}$ and it follows from (E.32) that $[\![\sigma]\!]_{\mathsf{SE}} = \mathcal{X}$. Thus, putting $\pi = \tau$ finishes the proof.

As for the right-hand side of (E.29), we can see that $\equiv_{\mathsf{SU}} \not\preceq \equiv_{\mathsf{RR}}$ because the programs $\mathcal{P} = \{ \sim p \leftarrow p. \}$ and $\mathcal{Q} = \{ \leftarrow p. \}$ are RR-equivalent but not SU-equivalent.

Similarly, programs $\mathcal{P} = \{ p. \}$ and $\mathcal{Q} = \{ p., p \leftarrow q. \}$ are RMR-equivalent but not RR-equivalent, so it follows that $\equiv_{\mathsf{RR}} \not\preceq \equiv_{\mathsf{RMR}}$.

Next, to verify that $\equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{RE}}$ it suffices to observe that the programs $\mathcal{P} = \{ p., q. \}$ and $\mathcal{Q} = \{ p., q \leftarrow p. \}$ are RE-equivalent but not RMR-equivalent.

Finally, programs $\mathcal{P} = \{ \sim p. \}$ and $\mathcal{Q} = \{ \leftarrow p. \}$ are SE-equivalent but not RE-equivalent, proving that $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SE}}$.

(2) It follows from Corollary E.44 that we can instead prove that

$$\models_{\mathsf{SE}} \preceq \models_{\mathsf{SMR}} \preceq \models_{\mathsf{SR}} \preceq \models_{\mathsf{RR}} \qquad \text{and} \qquad \equiv_{\mathsf{RR}} \not\preceq \equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{SE}} \quad . \tag{E.33}$$

We first concentrate on the left-hand side of (E.33). To prove that $\models_{\mathsf{SE}} \preceq \models_{\mathsf{SMR}}$, take some programs $\mathcal{P}$, $\mathcal{Q}$ with $\mathcal{P} \models_{\mathsf{SMR}} \mathcal{Q}$ and put $\mathcal{P}^\tau = \mathcal{P} \cup \{ \tau \}$. It follows that

$$\forall \sigma \in \mathcal{Q} \, \exists \pi_\sigma \in \mathcal{P}^\tau : [\![\pi_\sigma]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}} \quad . \tag{E.34}$$

We need to prove that $[\![\mathcal{P}]\!]_{\mathsf{SE}} \subseteq [\![\mathcal{Q}]\!]_{\mathsf{SE}}$. Suppose that $X \in [\![\mathcal{P}]\!]_{\mathsf{SE}}$. Then for all $\pi \in \mathcal{P}$,

$$X \in [\![\pi]\!]_{\mathsf{SE}} \quad . \tag{E.35}$$

Take some $\sigma \in \mathcal{Q}$. Our goal is to show that $X \in [\![\sigma]\!]_{\mathsf{SE}}$. By (E.34) there exists some $\pi_\sigma \in \mathcal{P}^\tau$ such that $[\![\pi_\sigma]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}}$. If $\pi_\sigma = \tau$, then it immediately follows that $X \in \mathcal{X} = [\![\tau]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}}$. If $\pi_\sigma \in \mathcal{P}$, then $X \in [\![\pi_\sigma]\!]_{\mathsf{SE}}$ by (E.31), so that $X \in [\![\sigma]\!]_{\mathsf{SE}}$.

Our next goal is to show that $\models_{\mathsf{SMR}} \preceq \models_{\mathsf{SR}}$. This follows directly by the definitions of $\models_{\mathsf{SMR}}$ and $\models_{\mathsf{SR}}$.

To prove the final part of the left-hand side of (E.33), suppose that $\mathcal{P} \models_{\mathsf{RR}} \mathcal{Q}$. Then

$$\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{RE}} = [\![\sigma]\!]_{\mathsf{RE}}$$

and, due to Lemma E.46, we obtain that

$$\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{SE}} = [\![\sigma]\!]_{\mathsf{SE}} \quad .$$

Consequently, $\mathcal{P} \models_{\mathsf{SR}} \mathcal{Q}$.

As for the right-hand side of (E.29), we can see that $\equiv_{\mathsf{RR}} \not\preceq \equiv_{\mathsf{SR}}$ because the programs $\mathcal{P} = \{ \sim p. \}$ and $\mathcal{Q} = \{ \leftarrow p. \}$ are SR-equivalent but not RR-equivalent.

Similarly, programs $\mathcal{P} = \{ p. \}$ and $\mathcal{Q} = \{ p., p \leftarrow q. \}$ are SMR-equivalent but not SR-equivalent, so it follows that $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{SMR}}$.

Finally, to verify that $\equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{SE}}$ it suffices to observe that the programs $\mathcal{P} = \{ p., q. \}$ and $\mathcal{Q} = \{ p., q \leftarrow p. \}$ are SE-equivalent but not SMR-equivalent.

(3) It follows from Corollary E.44 that we can instead prove that

$$\models_{\mathsf{SMR}} \preceq \models_{\mathsf{RMR}} \qquad \text{and} \qquad \equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{SMR}} \quad . \tag{E.36}$$

To show the former, take some programs $\mathcal{P}$, $\mathcal{Q}$ such that $\mathcal{P} \models_{\mathsf{RMR}} \mathcal{Q}$. It follows that

$$\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{RE}} \subseteq [\![\sigma]\!]_{\mathsf{RE}}$$

and, due to Lemma E.46, we obtain that

$$\forall \sigma \in \mathcal{Q} \, \exists \pi \in \mathcal{P} : [\![\pi]\!]_{\mathsf{SE}} \subseteq [\![\sigma]\!]_{\mathsf{SE}} \quad .$$

Consequently, $\mathcal{P} \models_{\mathsf{SMR}} \mathcal{Q}$.

As for the latter, it suffices to observe that the programs $\mathcal{P} = \{ \sim p. \}$ and $\mathcal{Q} = \{ \leftarrow p. \}$ are SMR-equivalent but not RMR-equivalent.

(4) According to Lemma E.43, it suffices to show that $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SMR}}$ and $\equiv_{\mathsf{SMR}} \not\preceq \equiv_{\mathsf{RE}}$. The former follows from the fact that the programs $\mathcal{P} = \{ \sim p. \}$ and $\mathcal{Q} = \{ \leftarrow p. \}$ are SMR-equivalent but not RE-equivalent. The latter can be verified by observing that though the programs $\mathcal{P} = \{ p., q. \}$ and $\mathcal{Q} = \{ p., q \leftarrow p. \}$ are RE-equivalent, they are not SMR-equivalent.

(5) According to Lemma E.43, it suffices to show that $\equiv_{\mathsf{RE}} \not\preceq \equiv_{\mathsf{SR}}$ and $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RE}}$. The former follows from the fact that the programs $\mathcal{P} = \{ \sim p. \}$ and $\mathcal{Q} = \{ \leftarrow p. \}$ are SR-equivalent but not RE-equivalent. The latter can be verified by observing that though the programs $\mathcal{P} = \{ p., q. \}$ and $\mathcal{Q} = \{ p., q \leftarrow p. \}$ are RE-equivalent, they are not SR-equivalent.

(6) According to Lemma E.43, it suffices to show that $\equiv_{\mathsf{RMR}} \not\preceq \equiv_{\mathsf{SR}}$ and $\equiv_{\mathsf{SR}} \not\preceq \equiv_{\mathsf{RMR}}$. The former follows from the fact that the programs $\mathcal{P} = \{ \sim p. \}$ and $\mathcal{Q} = \{ \leftarrow p. \}$ are SR-equivalent but not RMR-equivalent. The latter can be verified by observing that the programs $\mathcal{P} = \{ p. \}$ and $\mathcal{Q} = \{ p., p \leftarrow q. \}$ are RMR-equivalent, they are not SR-equivalent. □

F

Proofs: Exception-Based Updates

In the following we present proofs of results from Chapter 8, implicitly working under the same assumptions as those imposed in that chapter. That is, we constrain ourselves to propositional logic programs without explicit negation over a finite set of propositional atoms $\mathcal{A}$.

F.1 Exception-Based Rule Update Operators

Before we start with the proofs of syntactic and semantic properties of exception-based rule update operators, we take a closer look in Section F.1.1 at the conditions under which a set of three-valued interpretations forces an atom to have a certain truth value ($\mathcal{M}^J(p) = \mathsf{V}$), and under which two sets of three-valued interpretations are in a conflict ($\mathcal{M} \bowtie_p^J \mathcal{N}$). Then, in Sections F.1.2, F.1.3 and F.1.4, we prove the syntactic properties of δ_{a}-, δ_{b}- and δ_{c}-based rule update operators, respectively. Finally, Section F.1.5 is concerned with the semantic properties of exception-based rule update operators.

F.1.1 Conflict Between Sets of RE-Models

First we define some notation that we will need in the following.

Definition F.1 (Additional Notation)**.** Let X be a three-valued interpretation. Given an atom p, we say that X is an *RE-model of p* if $X(p) = \mathsf{T}$. We say that X is an *RE-model of $\sim p$* if $X(p) = \mathsf{F}$. We denote the set of all RE-models of a literal L by $[\![L]\!]_{\mathsf{RE}}$. Given a set of literals S, we say that X is an *RE-model of S* if X is an RE-model of all literals in S. We denote the set of all RE-models of S by $[\![S]\!]_{\mathsf{RE}}$.

Given a rule π, we say $H(\pi)^+$ is the *positive head of π*, $H(\pi)^-$ is the *negative head of π*, $B(\pi)^+$ is the *positive body of π* and $B(\pi)^-$ is the *negative body of π*.

Given a sequence of rule bases $\mathcal{R} = \langle \mathcal{R}_i \rangle_{i<n}$, we define

$$\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}} = \langle \langle\!\langle \mathcal{R}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{i<n} \enspace .$$

Lemma F.2. *Let π be a rule and J_1, J_2 be interpretations such that both π^{J_1} and π^{J_2} are different from τ. Then $\pi^{J_1} = \pi^{J_2}$.*

Proof. Follows directly from the definition of a rule reduct. □

Lemma F.3. *Let π be a rule, p an atom and $X = (I, J)$ a three-valued interpretation. If $(I \setminus \{ p \}, J \cup \{ p \})$ is not an RE-model of π, then the following holds:*

1. *Neither p nor $\sim p$ belongs to $B(\pi)$;*
2. *X is an RE-model of $B(\pi)$;*
3. *X is not an RE-model of any literal from $H(\pi) \setminus \{ p, \sim p \}$.*

Proof. From the assumption it follows that $\pi^{J \cup \{ p \}}$ is different from τ. This has two consequences. First, $H(\pi)^-$ is included in $J \cup \{ p \}$, so all atoms from $H(\pi)^-$, except possibly p, belong to J, and thus

$$X \text{ is not an RE-model of any default literal from } H(\pi) \setminus \{ \sim p \}. \tag{F.1}$$

The second consequence is that $B(\pi)^- \cap (J \cup \{ p \})$ is empty. Hence, $\sim p$ does not belong to $B(\pi)$. Furthermore, $B(\pi)^- \cap J$ must also be empty, so we can conclude that

$$X \text{ is an RE-model of all default literals from } B(\pi). \tag{F.2}$$

It also follows from the assumption that $I \setminus \{ p \}$ contains $B(\pi)^+$ but does not contain any atom from $H(\pi)^+$. As a consequence, p does not belong to $B(\pi)$ and we can also conclude that

$$X \text{ is an RE-model of all atoms from } B(\pi)\text{; and} \tag{F.3}$$

$$X \text{ is not an RE-model of any atom from } H(\pi) \setminus \{ p \}. \tag{F.4}$$

We can now use (F.2) and (F.3) to conclude that X is an RE-model of $B(\pi)$ and, similarly, we can use (F.1) and (F.4) to conclude that X is not an RE-model of any literal from $H(\pi) \setminus \{ p, \sim p \}$. □

Lemma F.4. *Let π be a rule, p an atom and $X = (I, J)$ a three-valued interpretation. Then $(I \setminus \{ p \}, J \cup \{ p \})$ is not an RE-model of π if the following holds:*

1. *Neither p nor $\sim p$ belongs to $B(\pi)$;*
2. *X is an RE-model of $B(\pi)$;*
3. *X is not an RE-model of any literal from $H(\pi) \setminus \{ p, \sim p \}$.*

Proof. We need to prove that $I \setminus \{ p \}$ is not a model of $\pi^{J \cup \{ p \}}$. We first need to show that $\pi^{J \cup \{ p \}}$ is equal to the rule $H(\pi)^+ \leftarrow B(\pi)^+$. This holds if $H(\pi)^-$ is included in $J \cup \{ p \}$ and $B(\pi)^-$ is disjoint with $J \cup \{ p \}$. Since X is an RE-model of $B(\pi)$, we can conclude that the set $B(\pi)^-$ is disjoint with J which, together with the assumption that $\sim p$ does not belong to $B(\pi)$, implies that $B(\pi)^-$ is disjoint with $J \cup \{ p \}$. We also know that X is not an RE-model of any literal from $H(\pi) \setminus \{ p, \sim p \}$, so we can conclude that $H(\pi)^- \setminus \{ p \}$ is included in J. Thus, $H(\pi)^-$ is included in $J \cup \{ p \}$ and we managed to prove that $\pi^{J \cup \{ p \}}$ is equal to the rule $H(\pi)^+ \leftarrow B(\pi)^+$.

It remains to show that $I \setminus \{ p \}$ includes $B(\pi)^+$ and that it does not contain any atom from $H(\pi)^+$. We know that X is an RE-model of $B(\pi)$, so I includes $B(\pi)^+$. Also, since

p does not belong to $B(\pi)$, $I \setminus \{ p \}$ also includes $B(\pi)^+$. Finally, we know that X is not an RE-model of any atom from $H(\pi)^+ \setminus \{ p \}$, so I does not contain any atom from $H(\pi)^+ \setminus \{ p \}$. This implies that $I \setminus \{ p \}$ does not contain any atom from $H(\pi)^+$. □

Proposition F.5. *Let π be a rule, p an atom and $X = (I, J)$ a three-valued interpretation. Then $(I \setminus \{ p \}, J \cup \{ p \})$ is not an RE-model of π if and only if the following holds:*

1. *Neither p nor $\sim p$ belongs to $B(\pi)$;*
2. *X is an RE-model of $B(\pi)$;*
3. *X is not an RE-model of any literal from $H(\pi) \setminus \{ p, \sim p \}$.*

Proof. Follows from Lemmas F.3 and F.4. □

Corollary F.6. *Let π be a rule, p an atom and J an interpretation with $p \in J$. If (J, J) is an RE-model of π but $(J \setminus \{ p \}, J)$ is not, then $p \in H(\pi)$ and $J \models B(\pi)$.*

Proof. It follows immediately from Proposition F.5 that $J \models B(\pi)$. Furthermore, by the definition of RE-model, J is a model of π^J while $J \setminus \{ p \}$ is not. Hence J contains some atom from $H(\pi)$ that is not contained in $J \setminus \{ p \}$. This atom can only be p. □

Lemma F.7. *Let π be a rule, $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$, p an atom, J an interpretation and V a truth value. If $\mathcal{M}^J(p) = \mathsf{V}$, then the following holds:*

1. *Neither p nor $\sim p$ belongs to $B(\pi)$;*
2. *J is a model of $B(\pi)$;*
3. *J is not a model of any literal from $H(\pi) \setminus \{ p, \sim p \}$;*
4. *One of the following conditions holds:*
   (a) *V is T and $H(\pi) \cap \{ p, \sim p \} = \{ p \}$, or*
   (b) *V is F and $H(\pi) \cap \{ p, \sim p \} = \{ \sim p \}$.*

Proof. First assume that $\mathsf{V} = \mathsf{T}$. Then $(J \cup \{ p \}, J \cup \{ p \})$ is an RE-model of π although both $(J \setminus \{ p \}, J \cup \{ p \})$ and $(J \setminus \{ p \}, J \setminus \{ p \})$ are not. By Proposition F.5 and Lemma E.45 we can conclude that the first three of the properties that we need to prove are satisfied. It remains to show that $H(\pi) \cap \{ p, \sim p \} = \{ p \}$, i.e. that p belongs to $H(\pi)^+$ but it does not belong to $H(\pi)^-$. To see that the former holds, note that $J \setminus \{ p \}$ is not a model of $\pi^{J \cup \{ p \}}$, so $J \setminus \{ p \}$ includes $B(\pi)^+$ and it does not contain any atom from $H(\pi)^+$. Since we know that $J \cup \{ p \}$ *is* a model of $\pi^{J \cup \{ p \}}$, it must be the case that $J \cup \{ p \}$ contains an atom from $H(\pi)^+$. This atom can only be p. Finally, if p were a member of $H(\pi)^-$, then $\pi^{J \setminus \{ p \}}$ would coincide with τ, so $(J \setminus \{ p \}, J \setminus \{ p \})$ would be an RE-model of π, contrary to the assumption.

Now assume that $\mathsf{V} = \mathsf{F}$. Then $(J \setminus \{ p \}, J \setminus \{ p \})$ is an RE-model of π although both $(J \setminus \{ p \}, J \cup \{ p \})$ and $(J \cup \{ p \}, J \cup \{ p \})$ are not. By Proposition F.5 and Lemma E.45 we can conclude that the first three of the properties that we need to prove are satisfied. It remains to show that $H(\pi) \cap \{ p, \sim p \} = \{ \sim p \}$, i.e. that p belongs to $H(\pi)^-$ but it does not belong to $H(\pi)^+$. To see that the former holds, note that by the assumption $J \setminus \{ p \}$ is a model of $\pi^{J \setminus \{ p \}}$ while it is not a model of $\pi^{J \cup \{ p \}}$. Hence, $\pi^{J \setminus \{ p \}}$ must be equal to τ. We know that $B(\pi)^-$ is disjoint with $J \cup \{ p \}$, so it must also be disjoint with $J \setminus \{ p \}$. Thus, there must exist some atom from $H(\pi)^-$ that is not contained in $J \setminus \{ p \}$ while it was contained in $J \cup \{ p \}$. This atom can only be p. Finally, if p were a member of $H(\pi)^+$, then $J \cup \{ p \}$ would contain an atom from $H(\pi)^+$, so $(J \cup \{ p \}, J \cup \{ p \})$ would be an RE-model of π, contrary to the assumption.

Finally, we show by contradiction that V cannot be equal to U. Suppose that V = U. Then the following conditions are satisfied:

$$(J \cup \{ p \}) \not\models \pi^{J \cup \{ p \}} \ , \tag{F.5}$$

$$(J \setminus \{ p \}) \models \pi^{J \cup \{ p \}} \ , \tag{F.6}$$

$$(J \setminus \{ p \}) \not\models \pi^{J \setminus \{ p \}} \ . \tag{F.7}$$

From (F.5) and (F.7) it follows that both $\pi^{J \setminus \{ p \}}$ and $\pi^{J \cup \{ p \}}$ are different from τ, so they must be identical (cf. Lemma F.2) and so (F.6) is in conflict with (F.7). □

Lemma F.8. *Let π be a rule, $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$, p an atom, J an interpretation and* V *a truth value. Then $\mathcal{M}^J(p) =$ V if the following holds:*

1. *Neither p nor $\sim p$ belongs to $B(\pi)$;*
2. *J is a model of $B(\pi)$;*
3. *J is not a model of any literal from $H(\pi) \setminus \{ p, \sim p \}$;*
4. *One of the following conditions holds:*
   (a) V *is* T *and $H(\pi) \cap \{ p, \sim p \} = \{ p \}$, or*
   (b) V *is* F *and $H(\pi) \cap \{ p, \sim p \} = \{ \sim p \}$.*

Proof. Let

$$X^{\mathsf{T}} = (J \cup \{ p \}, J \cup \{ p \}) \ ,$$
$$X^{\mathsf{U}} = (J \setminus \{ p \}, J \cup \{ p \}) \ ,$$
$$X^{\mathsf{F}} = (J \setminus \{ p \}, J \setminus \{ p \}) \ .$$

First suppose that V is T and $H(\pi) \cap \{ p, \sim p \} = \{ p \}$. We need to show that X^{T} is an RE-model of π while both X^{U} and X^{F} are not. The first property follows directly from the fact that p belongs to $H(\pi)^+$ and X^{T} is an RE-model of p. The second property follows from Proposition F.5 and Lemma E.45. To show that the third is also satisfied, note that since X^{U} is not an RE-model of π, the rule $\pi^{J \cup \{ p \}}$ coincides with the rule $H(\pi)^+ \leftarrow B(\pi)^+$. This implies that $B(\pi)^-$ is disjoint with $J \cup \{ p \}$ and $H(\pi)^-$ is included in $J \cup \{ p \}$. As a consequence, $B(\pi)^-$ is also disjoint with $J \setminus \{ p \}$. Moreover, from our assumptions we know that $H(\pi) \cap \{ p, \sim p \} = \{ p \}$, which means that p does not belong to $H(\pi)^-$. Thus, $H(\pi)^-$ is included in $J \setminus \{ p \}$. As a consequence, the rule $\pi^{J \setminus \{ p \}}$ also coincides with the rule $H(\pi)^+ \leftarrow B(\pi)^+$. Furthermore, since X^{U} is not an RE-model of π, $J \setminus \{ p \}$ is not a model of $\pi^{J \cup \{ p \}}$. Since $\pi^{J \cup \{ p \}} = \pi^{J \setminus \{ p \}}$, we obtain that $J \setminus \{ p \}$ is not a model of $\pi^{J \setminus \{ p \}}$. Hence X^{F} is not an RE-model of π.

Next, suppose that V is F and $H(\pi) \cap \{ p, \sim p \} = \{ \sim p \}$. We need to show that X^{F} is an RE-model of π while both X^{U} and X^{T} are not. The first property follows directly from the fact that p belongs to $H(\pi)^-$ but does not belong to $J \setminus \{ p \}$ because in this case $\pi^{J \setminus \{ p \}}$ coincides with τ. The second property follows from Proposition F.5 and Lemma E.45. To show that the third is also satisfied, note that since X^{U} is not an RE-model of π, the rule $\pi^{J \cup \{ p \}}$ coincides with the rule $H(\pi)^+ \leftarrow B(\pi)^+$ and $J \setminus \{ p \}$ is not a model of $\pi^{J \cup \{ p \}}$, i.e. $J \setminus \{ p \}$ includes $B(\pi)^+$ but does not contain any atom from $H(\pi)^+$. Thus, $J \cup \{ p \}$ also includes $B(\pi)^+$ and from our assumption that $H(\pi) \cap \{ p, \sim p \} = \{ \sim p \}$ we can conclude that p does not belong to $H(\pi)^+$. Thus, $J \cup \{ p \}$ does not contain any atom from $H(\pi)^+$ and, consequently, X^{T} is not an RE-model of π. □

Proposition F.9. *Let π be a rule, $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$, p an atom, J an interpretation and V a truth value. Then $\mathcal{M}^J(p) = \mathsf{V}$ if and only if the following holds:*

1. *Neither p nor $\sim p$ belongs to $B(\pi)$;*
2. *J is a model of $B(\pi)$;*
3. *J is not a model of any literal from $H(\pi) \setminus \{ p, \sim p \}$;*
4. *One of the following conditions holds:*
   (a) *V is T and $H(\pi) \cap \{ p, \sim p \} = \{ p \}$, or*
   (b) *V is F and $H(\pi) \cap \{ p, \sim p \} = \{ \sim p \}$.*

Proof. Follows from Lemmas F.7 and F.8. □

Proposition F.10. *Let π and σ be non-disjunctive rules, $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$, $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$, and J an interpretation. Then $\mathcal{M} \bowtie_p^J \mathcal{N}$ if and only if for some $L \in \{ p, \sim p \}$, $H(\pi) = \{ L \}$, $H(\sigma) = \{ \sim L \}$, J is a model of both $B(\pi)$ and $B(\sigma)$, and $B(\pi)$, $B(\sigma)$ do not contain p nor $\sim p$.*

Proof. Follows directly from Proposition F.9. □

Corollary F.11. *Let π and σ be facts, $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$, $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$ and J an interpretation. Then $\mathcal{M} \bowtie_p^J \mathcal{N}$ if and only if for some $L \in \{ p, \sim p \}$, $\pi = (L.)$ and $\sigma = (\sim L.)$.*

Proof. Follows directly from Proposition F.10. □

F.1.2 Syntactic Properties of δ_{a}-Based Operators

Definition F.12. Let $\mathcal{M}$ be a set of three-valued interpretations, $\mathcal{S}$ a set of sets of three-valued interpretations and δ a local exception function. We define

$$\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \mathcal{S}} \delta(\mathcal{M}, \mathcal{N}) \ .$$

We extend this definition to sequences of sets of sets of three-valued interpretations inductively as follows:

$$\mathsf{aug}_\delta(\mathcal{M}, \langle\rangle) = \mathcal{M} \ ,$$
$$\mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n+1}) = \mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n}), \mathcal{S}_n) \ .$$

Lemma F.13. *Let $\mathcal{M}$ be a set of three-valued interpretations, $\mathcal{S}$ a set of sets of three-valued interpretations and δ a local exception function. Then,*

$$\mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S} \rangle) = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) \ .$$

Proof. It suffices to observe that

$$\mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S} \rangle) = \mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \langle\rangle), \mathcal{S}) = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}) \ .$$ □

Proposition F.14. *Let $\mathcal{R} = \langle \mathcal{R}_i \rangle_{i<n}$ be a sequence of rule bases, $\mathcal{S}_i = \langle\!\langle \mathcal{R}_i \rangle\!\rangle_{\mathsf{RE}}$ for all $i < n$, and $\oplus$ a δ-based rule update operator. Then $\langle\!\langle \bigoplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ coincides with*

$$\{ \mathsf{aug}_\delta (\mathcal{M}, \langle \mathcal{S}_i \rangle_{j<i<n}) \mid \exists j < n : \mathcal{M} \in \mathcal{S}_j \} \ .$$

Proof. We prove by induction on n.

1° If $n = 1$, then $\langle\!\langle \bigoplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{R}_0 \rangle\!\rangle_{\mathsf{RE}}$ and

$$\begin{aligned}\{\, \mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_i \rangle_{j<i<n}) \mid \exists j < n : \mathcal{M} \in \mathcal{S}_j \,\} &= \{\, \mathsf{aug}_\delta(\mathcal{M}, \langle\rangle) \mid \mathcal{M} \in \mathcal{S}_0 \,\} \\ &= \{\, \mathcal{M} \mid \mathcal{M} \in \langle\!\langle \mathcal{R}_0 \rangle\!\rangle_{\mathsf{RE}} \,\} = \langle\!\langle \mathcal{R}_0 \rangle\!\rangle_{\mathsf{RE}} \enspace .\end{aligned}$$

2° Assuming that the proposition holds for n, we prove it for $n+1$. Let $\mathcal{R} = \langle \mathcal{R}_i \rangle_{i<n}$. By the inductive assumption we know that $\langle\!\langle \bigoplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ coincides with the set

$$\{\, \mathsf{aug}_\delta\,(\mathcal{M}, \langle \mathcal{S}_i \rangle_{j<i<n}) \mid \exists j < n : \mathcal{M} \in \mathcal{S}_j \,\} \enspace .$$

Let $\mathcal{R}' = \langle \mathcal{R}_i \rangle_{i \leq n}$. By the definition of $\bigoplus$, $\langle\!\langle \bigoplus \mathcal{R}' \rangle\!\rangle_{\mathsf{RE}}$ can be written as $\langle\!\langle (\bigoplus \mathcal{R}) \oplus \mathcal{R}_n \rangle\!\rangle_{\mathsf{RE}}$, expanded to

$$\left\{ \mathcal{M} \cup \bigcup_{\mathcal{N} \in \mathcal{S}_n} \delta(\mathcal{M}, \mathcal{N}) \;\middle|\; \mathcal{M} \in \left\langle\!\!\left\langle \bigoplus \mathcal{R}' \right\rangle\!\!\right\rangle_{\mathsf{RE}} \right\} \cup \langle\!\langle \mathcal{R}_n \rangle\!\rangle_{\mathsf{RE}}$$

and then simplified to

$$\left\{ \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}_n) \;\middle|\; \mathcal{M} \in \left\langle\!\!\left\langle \bigoplus \mathcal{R}' \right\rangle\!\!\right\rangle_{\mathsf{RE}} \right\} \cup \langle\!\langle \mathcal{R}_n \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

By using the inductive hypothesis and the definition of $\mathsf{aug}_\delta(\cdot, \cdot)$ we can also write this as

$$\{\, \mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_i \rangle_{j<i \leq n}) \mid \exists j < n : \mathcal{M} \in \mathcal{S}_j \,\} \cup \{\, \mathsf{aug}_\delta(\mathcal{M}, \langle\rangle) \mid \mathcal{M} \in \mathcal{S}_n \,\}$$

and then simplify it to

$$\{\, \mathsf{aug}_\delta(\mathcal{M}, \langle \mathcal{S}_i \rangle_{j<i \leq n}) \mid \exists j \leq n : \mathcal{M} \in \mathcal{S}_j \,\} \enspace . \qquad \square$$

Corollary F.15. *Let $\mathcal{R} = \langle \mathcal{R}_i \rangle_{i<n}$ be a sequence of rule bases and $\oplus$ a δ-based rule update operator. Then $\langle\!\langle \bigoplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ coincides with*

$$\{\, \mathsf{aug}_\delta\,([\![\pi]\!]_{\mathsf{RE}}, \langle \langle\!\langle \mathcal{R}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i<n}) \mid \exists j < n : \pi \in \mathcal{R}_j \,\} \enspace .$$

Proof. Follows from Proposition F.14 and from the fact that $[\![\cdot]\!]_{\mathsf{RE}}$ and $\mathsf{aug}_\delta(\cdot, \cdot)$ are functions. □

Lemma F.16. *Let $\mathcal{M}$ be a set of three-valued interpretations, $\mathcal{S}$ a sequence of sets of sets of three-valued interpretations, J an interpretation and p an atom. If (J, J) belongs to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \mathcal{S})$, but $(J \setminus \{\, p \,\}, J)$ does not, then (J, J) belongs to $\mathcal{M}$.*

Proof. Let $\mathcal{S} = \langle \mathcal{S}_i \rangle_{i<n}$. We prove by induction on n.

1° If $n = 0$, then the property follows trivially from the fact that $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \mathcal{S}) = \mathcal{M}$.

2° Suppose that the property holds for n. We will prove it for $n + 1$. So suppose that (J, J) belongs to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n+1})$ but $(J \setminus \{\, p \,\}, J)$ does not. Let $\mathcal{M}' = \mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n})$ and note that

$$\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n+1}) = \mathsf{aug}_{\delta_\mathsf{a}}(\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n}), \mathcal{S}_n) = \mathcal{M}' \cup \bigcup_{\mathcal{N} \in \mathcal{S}_n} \delta_\mathsf{a}(\mathcal{M}', \mathcal{N}) \enspace .$$

Thus, either (J, J) belongs to $\mathcal{M}'$ or to $\delta_\mathsf{a}(\mathcal{M}', \mathcal{N})$ for some $\mathcal{N} \in \mathcal{S}_n$. In the former case we can use the inductive assumption to conclude that (J, J) belongs to $\mathcal{M}$. In the latter case it follows from the definition of δ_a that $\delta_\mathsf{a}(\mathcal{M}', \mathcal{N})$ also contains $(J \setminus \{ p \}, J)$. But this is in conflict with the assumption that $(J, \setminus \{ p \}, J)$ does not belong to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle \mathcal{S}_i \rangle_{i<n+1})$. □

Proposition F.17. *Every δ_a-based rule update operator respects support.*

Proof. Let $\oplus$ be some δ_a-based rule update operator, pick some DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, suppose that J is a stable model of $\bigoplus \boldsymbol{P}$ and take some $p \in J$. We need to show that for some rule $\pi \in \mathsf{all}(\boldsymbol{P})$, $p \in H(\pi)$ and $J \models B(\pi)$.

Since J is a stable model of $\bigoplus \boldsymbol{P}$, we know that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$ and for all $I \subsetneq J$, (I, J) does not belong to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$. In particular, $(J \setminus \{ p \}, J)$ does not belong to $[\![\bigoplus \boldsymbol{P}]\!]_\mathsf{RE}$. Consequently, there is some set of three-valued interpretations $\mathcal{N} \in \langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_\mathsf{RE}$ such that $(J \setminus \{ p \}, J)$ does not belong to $\mathcal{N}$ although (J, J) does. According to Proposition F.14, $\mathcal{N}$ is of the form

$$\mathsf{aug}_\delta \left(\mathcal{M}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_\mathsf{RE} \rangle_{j<i<n} \right)$$

where $\mathcal{M} \in \langle\!\langle \mathcal{P}_j \rangle\!\rangle_\mathsf{RE}$ for some $j < n$. Let π be a rule from $\mathcal{P}_j$ such that $[\![\pi]\!]_\mathsf{RE} = \mathcal{M}$. Since $(J \setminus \{ p \}, J)$ does not belong to $\mathcal{N}$, it cannot belong to $\mathcal{M}$ since $\mathcal{M}$ is a subset of $\mathcal{N}$. Also, by Lemma F.16, (J, J) belongs to $\mathcal{M}$. It now follows from Corollary F.6 that $p \in H(\pi)$ and $J \models B(\pi)$. □

Lemma F.18. *Let $\boldsymbol{P}$ be a finite sequence of sets of facts and L a literal. Then,*

$$\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P} \rangle\!\rangle_\mathsf{RE}) = \begin{cases} \mathcal{X} & (\sim L.) \in \mathsf{all}(\boldsymbol{P}) \enspace ; \\ [\![L.]\!]_\mathsf{RE} & \textit{otherwise} \enspace . \end{cases}$$

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$. We prove by induction on n.

1° If $n = 0$, then the property follows trivially from the fact that $\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P} \rangle\!\rangle_\mathsf{RE}) = [\![L.]\!]_\mathsf{RE}$.

2° Suppose that the property holds for n; we prove it for $n + 1$. Put $\boldsymbol{P}' = \langle \mathcal{P}_i \rangle_{i \leq n}$ and note that $\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P}' \rangle\!\rangle_\mathsf{RE})$ is the same as

$$\mathsf{aug}_{\delta_\mathsf{a}}(\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P} \rangle\!\rangle_\mathsf{RE}), \langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE}) \enspace .$$

By the inductive assumption, one of the following cases occurs:

a) $(\sim L.) \in \mathsf{all}(\boldsymbol{P})$ and $\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P}' \rangle\!\rangle_\mathsf{RE}) = \mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{X}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE})$. Thus, $(\sim L.)$ also belongs to $\mathsf{all}(\boldsymbol{P}')$ and since $\mathcal{X}$ is a subset of $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{X}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE})$, we can conclude that $\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P}' \rangle\!\rangle_\mathsf{RE})$ is equal to $\mathcal{X}$.

b) $(\sim L.) \notin \mathsf{all}(\boldsymbol{P})$ and $\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P}' \rangle\!\rangle_\mathsf{RE}) = \mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE})$ which is the same as

$$[\![L.]\!]_\mathsf{RE} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE}} \delta_\mathsf{a}([\![L.]\!]_\mathsf{RE}, \mathcal{N}) \enspace .$$

Pick some $\mathcal{N} \in \langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE}$. By the definition of δ_a and by Corollary F.11 we can conclude that $\delta_\mathsf{a}([\![L.]\!]_\mathsf{RE}, \mathcal{N})$ is equal to $\mathcal{X}$ if and only if $\mathcal{N} = [\![\sim L.]\!]_\mathsf{RE}$; otherwise it is empty. Thus, $\mathsf{aug}_{\delta_\mathsf{a}}([\![L.]\!]_\mathsf{RE}, \langle\!\langle \boldsymbol{P}' \rangle\!\rangle_\mathsf{RE})$ is equal to $\mathcal{X}$ if and only if $[\![\sim L.]\!]_\mathsf{RE}$ belongs to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE}$; otherwise it is equal to $[\![L.]\!]_\mathsf{RE}$. It only remains to observe that since $(\sim L.)$ does not belong to $\mathsf{all}(\boldsymbol{P})$, it belongs to $\mathsf{all}(\boldsymbol{P}')$ if and only if it belongs to $\mathcal{P}_n$ which is if and only if $[\![\sim L.]\!]_\mathsf{RE}$ belongs to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_\mathsf{RE}$ due to the fact that two different facts are RE-canonical and thus not RE-equivalent. □

Corollary F.19. *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a finite sequence of sets of facts and $\oplus$ a δ_{a}-based rule update operator. Then $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \cup \{ \mathcal{X} \}$ coincides with*

$$\{ [\![L.]\!]_{\mathsf{RE}} \mid \exists j < n : (L.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim L.) \notin \mathcal{P}_i) \} \cup \{ \mathcal{X} \} \enspace .$$

Proof. By Corollary F.15, $\langle\!\langle \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ coincides with

$$\{ \mathsf{aug}_\delta ([\![\pi]\!]_{\mathsf{RE}}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i<n}) \mid \exists j < n : \pi \in \mathcal{P}_j \}$$

which can also be written as

$$\{ \mathsf{aug}_\delta ([\![L.]\!]_{\mathsf{RE}}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i<n}) \mid \exists j < n : (L.) \in \mathcal{P}_j \} \enspace .$$

Furthermore, due to Lemma F.18, we can equivalently write this as

$$\{ \mathcal{X} \mid \exists i, j, L : j < i < n \land (L.) \in \mathcal{P}_j \land (\sim L.) \in \mathcal{P}_i \} \\ \cup \{ [\![L.]\!]_{\mathsf{RE}} \mid \exists j < n : (L.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim L.) \notin \mathcal{P}_i) \} \enspace .$$

Thus, $\langle\!\langle \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \cup \{ \mathcal{X} \}$ is the same as

$$\{ [\![L.]\!]_{\mathsf{RE}} \mid \exists j < n : (L.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim L.) \notin \mathcal{P}_i) \} \cup \{ \mathcal{X} \} \enspace . \qquad \square$$

Proposition F.20. *Every δ_{a}-based rule update operator respects fact update.*

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a finite sequence of consistent sets of facts, J the interpretation

$$\{ p \mid \exists j < n : (p.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim p.) \notin \mathcal{P}_i) \}$$

and $\oplus$ a δ_{a}-based rule update operator. We need to show that J is the unique stable model of $\bigoplus \boldsymbol{P}$.

We start by proving that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$. Pick some set of three-valued interpretations $\mathcal{M}$ from $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. By Corollary F.19 we know that $\mathcal{M}$ is either $\mathcal{X}$, or it is equal to $[\![L.]\!]_{\mathsf{RE}}$ where

$$\exists j < n : (L.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim L.) \notin \mathcal{P}_i) \enspace .$$

In the former case it trivially holds that (J, J) belongs to $\mathcal{M} = \mathcal{X}$. Now suppose that L is an atom p. Then, by its definition, J contains p, so (J, J) belongs to $[\![p.]\!]_{\mathsf{RE}} = \mathcal{M}$. On the other hand, if L is a default literal $\sim p$, then the fact $(p.)$ does not belong to $\mathcal{P}_j$ because $\mathcal{P}_j$ is consistent, and it also does not belong to $\mathcal{P}_i$ for any i with $j < i < n$. So p does not belong to J and, hence, (J, J) belongs to $[\![\sim p.]\!]_{\mathsf{RE}} = \mathcal{M}$.

Now suppose that (I, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ and take some $p \in J$. Then,

$$\exists j < n : (p.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim p.) \notin \mathcal{P}_i) \enspace ,$$

so, by Corollary F.19, $[\![p.]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Since (I, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, it must also belong to $[\![p.]\!]_{\mathsf{RE}}$. Thus, p belongs to I and as the choice of $p \in J$ was arbitrary, we can conclude that $I = J$. As a consequence, J is indeed a stable model of $\bigoplus \boldsymbol{P}$.

It remains to prove that J is the only stable model of $\bigoplus \boldsymbol{P}$. Suppose that J' is a stable model of $\bigoplus \boldsymbol{P}$ and take some $p \in J$. We will show that p belongs to J'. We know that

$$\exists j < n : (p.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim p.) \notin \mathcal{P}_i) \enspace ,$$

so, by Corollary F.19, $[\![p.]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Since J' is a stable model of $\bigoplus \boldsymbol{P}$, (J', J') belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ and, consequently, also to $[\![p.]\!]_{\mathsf{RE}}$. Thus, p must belong to J'. Now take some atom p that does not belong to J. We will show that p does not belong to J' either. There are two cases to consider:

a) If $(p.)$ does not belong to $\mathcal{P}_j$ for all $j < n$, then it can be seen that $(J' \setminus \{ p \}, J')$ belongs to all elements of $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Thus, since J' is a stable model of $\bigoplus \boldsymbol{P}$, $J' \setminus \{ p \} = J'$ and, consequently, p does not belong to J'.

b) If $(p.)$ belongs to $\mathcal{P}_{j_0}$ for some $j_0 < n$ and whenever $(p.)$ belongs to $\mathcal{P}_j$ for some j, there is some i with $j < i < n$ such that $(\sim p.)$ belongs to $\mathcal{P}_i$, then there must exist some j_1 such that $(\sim p.)$ belongs to $\mathcal{P}_{j_1}$ and for all i with $j_1 < i < n$, $(p.)$ does not belong to $\mathcal{P}_i$. Consequently, $[\![\sim p.]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. Thus, since (J', J') belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, it follows that p cannot belong to J'.

As desired, we have shown that $J = J'$. □

Proposition F.21. *Let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$ be a dynamic logic program, $\oplus$ a δ_{a}-based rule update operator and J an interpretation. If J is a stable model of $\bigoplus \boldsymbol{P}$, then J is a justified update model of $\boldsymbol{P}$.*

Proof. From the assumption we can conclude that (J, J) is an RE-model of $\mathcal{P} \oplus \mathcal{U}$ and for every $I \subsetneq J$, (I, J) is not an RE-model of $\mathcal{P} \oplus \mathcal{U}$.

We need to show that J is a minimal model of the program

$$\mathcal{Q} = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J \enspace .$$

First we prove that J is a model of $\mathcal{Q}$. Take some rule $\pi' \in \mathcal{Q}$ and let π be a rule from $[\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]$ such that $\pi' = \pi^J$. We consider two cases:

a) If π belongs to $\mathcal{U}$, then since (J, J) belongs to $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{RE}}$ and $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ contains $[\![\pi]\!]_{\mathsf{RE}}$, (J, J) must also belong to $[\![\pi]\!]_{\mathsf{RE}}$. Thus, J is a model of π and consequently also a model of $\pi' = \pi^J$.

b) If π belongs to $\mathcal{P} \setminus \mathsf{rej}(\boldsymbol{P}, J)$, then since (J, J) belongs to $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{RE}}$ and $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ contains $[\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$, (J, J) must also belong to

$$[\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \enspace .$$

Suppose first that for some rule $\sigma \in \mathcal{U}$, (J, J) belongs to the set

$$\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \enspace .$$

This implies that $[\![\pi]\!]_{\mathsf{RE}} \bowtie_p^J [\![\sigma]\!]_{\mathsf{RE}}$ for some atom p and by Proposition F.10 we can conclude that π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$, contrary to the assumption. Thus, (J, J) does not belong to the set

$$\bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$$

and, consequently, it belongs to $[\![\pi]\!]_{\mathsf{RE}}$. Hence, J is a model of π and, consequently, it is also a model of $\pi' = \pi^J$.

It remains to prove that J is a minimal model of $\mathcal{Q}$. Take some model I of $\mathcal{Q}$ such that I is a subset of J. We need to show that $I = J$.

In the following we will show that (I, J) is a member of the set $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{RE}}$ which, together with the assumption that J is a stable model of $\mathcal{P} \oplus \mathcal{U}$, implies that $I = J$.

So in order to finish the proof, take some set $\mathcal{M}$ from $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. We need to show that (I, J) belongs to $\mathcal{M}$. Recall that

$$\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \left\{ [\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \;\middle|\; \pi \in \mathcal{P} \right\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

If $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$, then there is some rule $\pi \in \mathcal{U}$ such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. Moreover, π^J belongs to $\mathcal{Q}$, so I is a model of π^J. It then follows that (I, J) is an RE-model of π, i.e. that (I, J) belongs to $\mathcal{M}$, as we wanted to show.

The remaining case is when for some $\pi \in \mathcal{P}$,

$$\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \enspace .$$

Suppose that (I, J) does not belong to $[\![\pi]\!]_{\mathsf{RE}}$. Then I is not a model of π^J. Since I is a subset of J, we can conclude from this that

$$J \models B(\pi) \enspace . \tag{F.8}$$

Furthermore, from our assumption that I is a model of $\mathcal{Q}$ it then follows that π^J does not belong to $\mathcal{Q}$ and, consequently, π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$. So there must be some rule $\sigma \in \mathcal{U}$ such that $H(\pi) = {\sim}H(\sigma)$ and $J \models B(\sigma)$. Since we know from the previous part of the proof that J is a model of $\mathcal{Q}$, we can conclude that $J \models H(\sigma)$, so $J \not\models H(\pi)$.

Thus, it follows from (F.8) that J is not a model of π, so (J, J) is not an RE-model of π. But since J is a stable model of $\mathcal{P} \oplus \mathcal{U}$, (J, J) must belong to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma']\!]_{\mathsf{RE}})$ for some $\sigma' \in \mathcal{U}$ and from the definition of $\delta_{\mathsf{a}}(\cdot, \cdot)$ we obtain that (I, J) also belongs to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma']\!]_{\mathsf{RE}})$. This implies that (I, J) belongs to $\mathcal{M}$ and our proof is finished. □

Proposition F.22. *Let $\boldsymbol{P} = \langle \mathcal{P}, \mathcal{U} \rangle$ be a DLP free of local cycles, $\oplus$ a δ_{a}-based rule update operator and J an interpretation. If J is a justified update model of $\boldsymbol{P}$, then J is a stable model of $\bigoplus \mathcal{P}$.*

Proof. Suppose that J is a justified update model of $\langle \mathcal{P}, \mathcal{U} \rangle$. Then it is a minimal model of the program

$$\mathcal{Q} = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J \enspace .$$

We need to prove that (J, J) is an RE-model of $\mathcal{P} \oplus \mathcal{U}$ and for every $I \subsetneq J$, (I, J) is not an RE-model of $\mathcal{P} \oplus \mathcal{U}$.

In order to show that (J, J) is an RE-model of $\mathcal{P} \oplus \mathcal{U}$, recall that

$$\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \left\{ [\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \;\middle|\; \pi \in \mathcal{P} \right\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

and take some set $\mathcal{M}$ from $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. If $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$, then there is a rule $\pi \in \mathcal{U}$ such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. Also, π^J belongs to $\mathcal{Q}$, so J is a model of π^J. Consequently, (J, J) belongs to $\mathcal{M}$.

Now suppose that for some π from $\mathcal{P}$,

$$\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{Q}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \enspace .$$

If (J, J) does not belong to $[\![\pi]\!]_{\mathsf{RE}}$, then J is not a model of π^J, so π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$. So there exists a rule σ from $\mathcal{U}$ such that $H(\pi) = {\sim}H(\sigma)$ and $J \models B(\sigma)$. The previous conclusions, together with the fact that π and σ are not local cycles, allow us to use Proposition F.10 and conclude that $[\![\pi]\!]_{\mathsf{RE}} \bowtie_p^J [\![\sigma]\!]_{\mathsf{RE}}$ holds for some atom p. Hence, (J, J) belongs to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$, and consequently also to $\mathcal{M}$.

Now suppose that (I, J) belongs to $[\![\pi]\!]_{\mathsf{RE}}$. We will show that I is a model of $\mathcal{Q}$, which implies that $I = J$ because J is by assumption a minimal model of $\mathcal{Q}$.

Take some rule π' from $\mathcal{Q}$ and suppose that $\pi' = \pi^J$ for some $\pi \in [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]$. If π belongs to $\mathcal{U}$, then $[\![\pi]\!]_{\mathsf{RE}}$ belongs to $\langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. Consequently, (I, J) belongs to $[\![\pi]\!]_{\mathsf{RE}}$, so I is a model of π'.

The final case to consider is when π belongs to $\mathcal{P}$. We will prove by contradiction that I is a model of π'. So suppose that I is not a model of π'. Then (I, J) is not an RE-model of π. However, since by assumption (I, J) belongs to $[\![\mathcal{P} \oplus \mathcal{U}]\!]_{\mathsf{RE}}$, it must also belong to the set

$$[\![\pi]\!]_{\mathsf{RE}} \cup \bigcup_{\sigma \in \mathcal{U}} \delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}}) \enspace .$$

We have already shown that it is not a member of $[\![\pi]\!]_{\mathsf{RE}}$, so there must exist some $\sigma \in \mathcal{U}$ such that (I, J) belongs to $\delta_{\mathsf{a}}([\![\pi]\!]_{\mathsf{RE}}, [\![\sigma]\!]_{\mathsf{RE}})$. Thus, $[\![\pi]\!]_{\mathsf{RE}} \bowtie_p^J [\![\sigma]\!]_{\mathsf{RE}}$ holds for some atom p. We can use Proposition F.10 to conclude that $H(\pi) = {\sim}H(\sigma)$ and $J \models B(\sigma)$. Hence, π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$, contrary to our assumption. □

Theorem 8.6. *Every δ_{a}-based rule update operator respects support and fact update. Furthermore, it also respects causal rejection and acyclic justified update w.r.t. DLPs of length at most two.*

Proof. Follows from Propositions F.17 and F.20, F.21 and F.22 and the fact that justified update models satisfy the latter two properties. □

F.1.3 Syntactic Properties of δ_{b}-Based Operators

Lemma F.23. *Let $\mathcal{M}$ be an RE-rule-expressible set of three-valued interpretations, $\mathcal{S}$ a set of RE-rule-expressible sets of three-valued interpretations, J an interpretation, p an atom and V_0 a truth value.*

If $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})^J(p) = \mathsf{V}_0$, then $\mathcal{M}^J(p) = \mathsf{V}_0$.

Proof. Suppose that $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})^J(p) = \mathsf{V}_0$. By the definition we then obtain that for all truth values V,

$$J[\mathsf{V}/p] \in \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S}) \text{ if and only if } \mathsf{V} = \mathsf{V}_0 \enspace . \tag{F.9}$$

If the interpretation $J[\mathsf{V}_0/p]$ belongs to $\mathcal{M}$, then we can use (F.9) together with the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})$ to conclude that $\mathcal{M}^J(p) = \mathsf{V}_0$ and our proof ends.

So suppose that $J[\mathsf{V}_0/p]$ does not belong to $\mathcal{M}$. Then it follows from (F.9) and from the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})$ that the interpretations $J[\mathsf{T}/p]$, $J[\mathsf{U}/p]$ and $J[\mathsf{F}/p]$ do not belong to $\mathcal{M}$. Thus, since $J[\mathsf{V}_0/p]$ belongs to $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})$, there must exist some $\mathcal{N} \in \mathcal{S}$ such that $J[\mathsf{V}_0/p] = (I, K)$ belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$. In other words, there exists an atom q and an interpretation J' such that $I \subseteq J' \subseteq K$ and $\mathcal{M}^{J'}(q) \neq \mathcal{N}^{J'}(q)$. Note that $J \setminus \{p\} \subseteq I \subseteq J' \subseteq K \subseteq J \cup \{p\}$. Thus, $(J', J') = J[\mathsf{V}_1/p]$ for some $\mathsf{V}_1 \in \{\mathsf{T}, \mathsf{F}\}$. We now distinguish three cases:

a) If $\mathsf{V}_1 = \mathsf{T}$, then by the definition of $\delta_{\mathsf{b}}(\cdot, \cdot)$ we obtain that both $J[\mathsf{T}/p]$ and $J[\mathsf{U}/p]$ belong to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$.

b) If $\mathsf{V}_1 = \mathsf{F}$ and $p \neq q$, then by the definition of $\delta_{\mathsf{b}}(\cdot, \cdot)$ we obtain that both $J[\mathsf{F}/p]$ and $J[\mathsf{U}/p]$ belong to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$.

c) If $\mathsf{V}_1 = \mathsf{F}$ and $p = q$, then by Proposition F.9 it follows that for $J'' = J' \cup \{ p \}$ we also have $\mathcal{M}^{J''}(p) \neq \mathcal{N}^{J''}(p)$. Thus, by the definition of $\delta_{\mathsf{b}}(\cdot, \cdot)$ we obtain that $J[\mathsf{F}/p]$, $J[\mathsf{U}/p]$ and $J[\mathsf{T}/p]$ belong to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$.

In either case, it is not possible that $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})^J(p)$ is defined, a conflict with our assumption. □

Proposition F.24 (Exception Independence for Rules)**.** *Let $\mathcal{M}$ be a set of three-valued interpretations that is* **RE***-rule-expressible by a non-disjunctive rule and $\mathcal{S}$, $\mathcal{T}$ be sets of* **RE***-rule-expressible sets of three-valued interpretations. Then the following holds:*

$$\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S}), \mathcal{T}) = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S} \cup \mathcal{T}) \ .$$

Proof. By applying the definition of $\mathsf{aug}_{\delta_{\mathsf{b}}}(\cdot, \cdot)$ we can see that our goal is to show that the set

$$\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S}) \cup \bigcup_{\mathcal{N} \in \mathcal{T}} \delta_{\mathsf{b}}(\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S}), \mathcal{N}) \tag{F.10}$$

is equal to the set $\mathcal{M} \cup \bigcup_{\mathcal{N} \in \mathcal{S} \cup \mathcal{T}} \delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$ which can also be written as

$$\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S}) \cup \bigcup_{\mathcal{N} \in \mathcal{T}} \delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N}) \ . \tag{F.11}$$

First suppose that some three-valued interpretation $X = (I, J)$ belongs to (F.10). If X belongs to $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})$, then it directly follows that X also belongs to (F.11). So suppose that X belongs to

$$\delta_{\mathsf{b}}(\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S}), \mathcal{N})$$

for some $\mathcal{N} \in \mathcal{T}$. By the definition of $\delta_{\mathsf{b}}(\cdot, \cdot)$ we obtain that there exists some atom p and some interpretation J with certain properties relative to X such that $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})^J(p) \neq \mathcal{N}^J(p)$. By Lemma F.23 we then conclude that $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})^J(p) = \mathcal{M}^J(p)$. Thus, X also belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$ and, consequently, also to the set (F.11).

Now suppose that some three-valued interpretation $X = (I, K)$ belongs to (F.11). The case when X belongs to $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})$ is trivial, so we assume that X belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$ for some $\mathcal{N} \in \mathcal{T}$. This implies that there exists an atom p and some interpretation J such that $I \subseteq J \subseteq K$ and $\mathcal{M}^J(p) \neq \mathcal{N}^J(p)$. Suppose that $\mathcal{M}^J(p) = \mathsf{V}_0$. If it also holds that $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})^J(p) = \mathsf{V}_0$, then it can be seen that X belongs to (F.10). Otherwise it follows from the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})$ that $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})^J(p)$ is undefined and it contains both $J[\mathsf{V}_0/p]$ and $J[\mathsf{V}_1/p] = (I', K')$ for some $\mathsf{V}_1 \neq \mathsf{V}_0$. Thus, for some $\mathcal{N}' \in \mathcal{S}$ it holds that (I', K') belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N}')$. In other words, there exists an atom q and an interpretation J' such that $I' \subseteq J' \subseteq K'$ and $\mathcal{M}^{J'}(q) \neq \mathcal{N}'^{J'}(q)$. Since $\mathcal{M}$ is expressible by a non-disjunctive rule, it follows from Proposition F.9 that $q = p$. Also, J and J' may only differ in the valuation of p, so we obtain that $\mathcal{M}^J(p) \neq \mathcal{N}'^J(p)$. Consequently, X belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N}')$, so it also belongs to (F.10). □

The previous proposition does not work when $\mathcal{M}$ is a disjunctive rule. For example, when $(p; q.)$ is updated by $\{ \sim p \leftarrow \sim r., \sim q \leftarrow \sim r. \}$, the interpretation $(\emptyset, qr)$ is introduced as an exception. However, that same interpretation is not introduced as an exception when $(p; q.)$ is updated first by $(\sim q \leftarrow \sim r.)$ and only then by $(\sim p \leftarrow \sim r.)$.

Definition F.25. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP. We define the set $\mathsf{after}_j(\boldsymbol{P})$ as

$$\mathsf{after}_j(\boldsymbol{P}) = \bigcup_{j<i<n} \mathcal{P}_i \ .$$

Proposition F.26 (Exception Independence for Programs)**.** *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP and $\oplus$ a δ_b-based rule update operator. Then $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ coincides with*

$$\left\{ \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}} \right) \;\middle|\; \exists j < n : \mathcal{M} \in \langle\!\langle \mathcal{P}_j \rangle\!\rangle_{\mathsf{RE}} \right\} \enspace .$$

Proof. It follows from Proposition F.14 that $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ coincides with

$$\left\{ \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i<n} \right) \;\middle|\; \exists j < n : \mathcal{M} \in \langle\!\langle \mathcal{P}_j \rangle\!\rangle_{\mathsf{RE}} \right\} \enspace .$$

Take some $\mathcal{M} \in \langle\!\langle \mathcal{P}_j \rangle\!\rangle_{\mathsf{RE}}$ for some $j < n$. We will prove by induction on n that

$$\mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i<n} \right) = \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}} \right) \enspace .$$

1° If $n = 1$,

$$\mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle \rangle \right) = \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \emptyset \right) = \mathcal{M} \enspace .$$

2° Put $\boldsymbol{P}' = \langle \mathcal{P}_i \rangle_{i \leq n}$ and suppose that the property holds for n, i.e.

$$\mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i<n} \right) = \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}} \right) \enspace .$$

Using Proposition F.26 we can now derive the property for $n + 1$:

$$\begin{aligned}
&\mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i\leq n} \right) \\
&\qquad = \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}} \rangle_{j<i<n} \right), \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}} \right) \\
&\qquad = \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}} \right), \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}} \right) \\
&\qquad = \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}} \cup \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}} \right) \\
&\qquad = \mathsf{aug}_{\delta_\mathsf{b}} \left(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}') \rangle\!\rangle_{\mathsf{RE}} \right) \enspace .
\end{aligned}$$

□

Proposition F.27. *Let $\boldsymbol{P}$ be a dynamic logic program, $\oplus$ a δ_b-based rule update operator and J an interpretation. If J is a stable model of $\bigoplus \boldsymbol{P}$, then J is a justified update model of $\boldsymbol{P}$.*

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$. From the assumption we can conclude that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ and for every $I \subsetneq J$, (I, J) does not belong to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$.

We need to show that J is a minimal model of the program

$$\mathcal{P}' = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J \enspace .$$

First we prove that J is a model of $\mathcal{P}'$. Take some rule $\pi' \in \mathcal{P}'$ and let π be a rule from $\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)$ such that $\pi' = \pi^J$. Then there is some $j < n$ such that π belongs to $\mathcal{P}_j$ and there is no index i and rule σ with $j < i < n$ and $\sigma \in \mathcal{P}_i$ such that $H(\pi) = {\sim}H(\sigma)$ and $J \models B(\sigma)$. Let $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. Since π belongs to $\mathcal{P}_j$, we know that $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ contains the set

$$\mathsf{aug}_{\delta_\mathsf{b}}(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}} \delta_\mathsf{b}(\mathcal{M}, \mathcal{N}) \enspace . \tag{F.12}$$

Furthermore, since (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, it must also belong to (F.12). If (J, J) belongs to $\mathcal{M}$, then J is a model of π, so it is also a model of π' as desired. So suppose that (J, J) does not belong to $\mathcal{M}$. Then for some i with $j < i < n$ there exists a member $\mathcal{N}$ of $\langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}}$ such that (J, J) belongs to $\delta_\mathsf{b}(\mathcal{M}, \mathcal{N})$. Thus, for some atom p, $\mathcal{M} \bowtie_p^J \mathcal{N}$, and by Proposition F.10 we conclude that there is a rule σ in $\mathcal{P}_i$ such that $H(\pi) = {\sim}H(\sigma)$ and $J \models B(\sigma)$, contrary to our previous assumption.

It remains to prove that J is a minimal model of $\mathcal{P}'$. Take some model I of $\mathcal{P}'$ such that

I is a subset of J. We need to show that $I = J$. In the following we will show that (I, J) is a member of $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ which, together with the assumption that J is a stable model of $\bigoplus \boldsymbol{P}$, implies that $I = J$. So in order to finish the proof, take some set

$$\mathsf{aug}_{\delta_\mathsf{b}}(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}} \delta_\mathsf{b}(\mathcal{M}, \mathcal{N}) \enspace . \tag{F.13}$$

from $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$ and π belongs to $\mathcal{P}_j$. We need to show that (I, J) belongs to (F.13). This obviously holds if (I, J) belongs to $\mathcal{M}$, so we can assume that (I, J) does not belong to $\mathcal{M}$. Then, I is not a model of π^J. Thus, π^J is different from τ and, consequently, $J \models B(\pi)^-$. Also, $B(\pi)^+$ is included in I but $H(\pi)^+$ is not. Since I is a subset of J, this implies that $B(\pi)^+$ is included in J, so

$$J \models B(\pi) \enspace . \tag{F.14}$$

Moreover, since I is a model of $\mathcal{P}'$, it follows that π^J does not belong to $\mathcal{P}'$, so π must belong to $\mathsf{rej}(\boldsymbol{P}, J)$. Thus, there must exist a rule $\sigma \in \mathcal{P}_i$ for some i with $j < i < n$ such that

$$H(\pi) = L \text{ and } H(\sigma) = {\sim}L \text{ and } J \models B(\sigma) \enspace . \tag{F.15}$$

Let $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$. If the bodies of π and σ do not contain L nor ${\sim}L$, then we can use Proposition F.10 to conclude that there is an atom p such that $\mathcal{M} \bowtie_p^J \mathcal{N}$. Thus, by the definition of $\delta_\mathsf{b}(\cdot, \cdot)$, (I, J) belongs to $\delta_\mathsf{b}(\mathcal{M}, \mathcal{N})$, which is a subset of $\mathsf{aug}_{\delta_\mathsf{b}}(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}})$. It only remains to consider the case when L or ${\sim}L$ belongs to the body of π or σ:

- If L belongs to the body of π, then we arrive at a conflict with the assumption that (I, J) is not an RE-model of π.

- If ${\sim}L$ belongs to the body of π, then it follows from (F.14) and (F.15) that (J, J) is not an RE-model of π. At the same time, $\delta_\mathsf{b}(\mathcal{M}, \mathcal{N})$ is empty for all $\mathcal{N}$ because for all interpretations J' and atoms q, it is impossible for $\mathcal{M}^{J'}(q)$ to be defined. Thus, we obtain a conflict with the assumption that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$.

- If L belongs to the body of σ, then it follows from (F.15) that (J, J) is not an RE-model of σ. At the same time, $\delta_\mathsf{b}(\mathcal{N}, \mathcal{N}')$ is empty for all $\mathcal{N}'$ because for all interpretations J' and atoms q, it is impossible for $\mathcal{N}^{J'}(q)$ to be defined. Thus, we obtain a conflict with the assumption that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$.

- Finally, if ${\sim}L$ belongs to the body of σ, then it follows from (F.15) that $J \models {\sim}L$ and together with (F.14) we obtain that J is not a model of π. Thus (J, J) is not an RE-model of π and since (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, there must exist some $\mathcal{N}'$ from $\langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}$ such that (J, J) belongs to $\delta_\mathsf{b}(\mathcal{M}, \mathcal{N}')$. By the definition of $\delta_\mathsf{b}(\cdot, \cdot)$ we then obtain that (I, J) also belongs to $\delta_\mathsf{b}(\mathcal{M}, \mathcal{N}')$, and thus it also belongs to $\mathsf{aug}_{\delta_\mathsf{b}}(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}})$. □

Proposition F.28. *Let $\boldsymbol{P}$ be a DLP free of local cycles, $\oplus$ a δ_b-based rule update operator and J an interpretation. If J is a justified update model of $\boldsymbol{P}$, then J is a stable model of $\bigoplus \mathcal{P}$.*

Proof. Suppose that J is a justified update model of $\boldsymbol{P}$. Then J is a minimal model of the program

$$\mathcal{P}' = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)]^J \enspace .$$

We need to prove that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$ and for every $I \subsetneq J$, (I, J) does not belong to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$.

In order to show that (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, take some set

$$\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}} \delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N}) \tag{F.16}$$

from $\langle\!\langle \bigoplus \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ such that $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$ and π belongs to $\mathcal{P}_j$. If (J, J) belongs to $\mathcal{M}$, then it obviously belongs to (F.16). So let's assume that (J, J) does not belong to $\mathcal{M}$. Then J is not a model of π. Thus,

$$J \models B(\pi) \enspace , \tag{F.17}$$

and we can conclude that π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$. As a consequence, there exists a rule σ from $\mathcal{P}_j$ for some $j > i$ such that

$$H(\pi) = L \text{ and } H(\sigma) = {\sim}L \text{ and } J \models B(\sigma) \enspace . \tag{F.18}$$

Let $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$. It can be verified that (F.17) and (F.18), together with the assumption that π and σ are not local cycles, allow us to use Proposition F.10 and conclude that for some atom p, $\mathcal{M} \bowtie_p^J \mathcal{N}$. Thus, (J, J) belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$, and consequently also to (F.16).

Now suppose that (I, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$. We will show that I is a model of $\mathcal{P}'$, which implies that $I = J$ because J is by assumption a minimal model of $\mathcal{P}'$. Take some rule π' from $\mathcal{P}'$. Then there is a rule π from $\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)$ such that $\pi' = \pi^J$. Suppose that π belongs to $\mathcal{P}_j$ and there is no rule $\sigma \in \mathcal{P}_i$ for some i with $j < i < n$, $H(\pi) = {\sim}H(\sigma)$ and $J \models B(\sigma)$.

We will prove by contradiction that I is a model of π'. So suppose that I is not a model of π'. Then π^J is different from τ and, consequently, $J \models B(\pi)^-$. Also, $B(\pi)^+$ is included in I, so since I is a subset of J, $B(\pi)^+$ is included in J as well. Hence,

$$J \models B(\pi) \enspace . \tag{F.19}$$

Also, (I, J) is not an RE-model of π. However, since by assumption (I, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, it must also belong to the set

$$\mathcal{M} \cup \bigcup_{\mathcal{N} \in \langle\!\langle \mathsf{after}_j(\boldsymbol{P}) \rangle\!\rangle_{\mathsf{RE}}} \delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$$

where $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$. We have already shown that it is not a member of $\mathcal{M}$, so there must a rule $\sigma \in \mathcal{P}_i$ for some i with $j < i < n$ such that (I, J) belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$ where $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$. Thus, there exists some interpretation K and an atom p such that $I \subseteq K \subseteq J$, $\mathcal{M} \bowtie_p^K \mathcal{N}$ and if p belongs to $J \setminus I$, then $J = K$. By Proposition F.10, there is a literal $L \in \{ p, {\sim}p \}$ such that $H(\pi) = \{ L \}$ and $H(\sigma) = \{ {\sim}L \}$. We now consider two cases:

- If $p \notin J \setminus I$, then $(I, J)(p) = (J, J)(p)$. If J is a model of π, then from (F.19) we obtain that J is a model of $H(\pi)$. Thus (J, J) is an RE-model of $H(\pi)$ and consequently (I, J) is also an RE-model of $H(\pi)$. But this is in conflict with the assumption that (I, J) is not an RE-model of π.

  The remaining case is when J is not a model of π. In this case, (J, J) does not belong to $\mathcal{M}$. Furthermore, since (J, J) belongs to $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{RE}}$, it also implies that there is a set $\mathcal{N}' \in \langle\!\langle \mathcal{P}_i \rangle\!\rangle_{\mathsf{RE}}$ for some i' with $j < i' < n$ such that (J, J) belongs to $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N}')$. Thus, $\mathcal{M} \bowtie_p^J \mathcal{N}'$, so, by Proposition F.10, π belongs to $\mathsf{rej}(\boldsymbol{P}, J)$, contrary to the way it was picked.

- If $p \in J \setminus I$, then $J = K$, so $\mathcal{M} \bowtie_p^J \mathcal{N}$. Thus, by Proposition F.10, π belongs to

$\mathsf{rej}(\boldsymbol{P}, J)$, contrary to the way it was picked. □

F.1.4 Syntactic Properties of δ_{c}-Based Operators

Lemma F.29. *Let $\boldsymbol{P}$ be a DLP, $\oplus_\beta$ a δ_{b}-based rule update operator and $\oplus_\gamma$ a δ_{c}-based rule update operator. If $\mathcal{M}$ belongs to $\langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$, then either $\mathcal{M} = \mathfrak{X}$, or $\mathcal{M}$ belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$.*

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$. We prove by induction on n:

1° If $n = 0$, then the condition is vacuously satisfied.

2° Assuming that the statement holds for n, we prove it for $n+1$. Let $\boldsymbol{P}' = \langle \mathcal{P}_i \rangle_{i \leq n}$ and suppose that $\mathcal{M}'$ belongs to $\langle\!\langle \bigoplus_\gamma \boldsymbol{P}' \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle (\bigoplus_\gamma \boldsymbol{P}) \oplus_\gamma \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, i.e. it belongs to

$$\left\{ \mathsf{aug}_{\delta_{\mathsf{c}}}(\mathcal{M}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}) \;\middle|\; \mathcal{M} \in \langle\!\langle \textstyle\bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \right\} \cup \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

If $\mathcal{M}'$ belongs to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, then it also belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P}' \rangle\!\rangle_{\mathsf{RE}}$ and the proof ends.
In the principal case we know that $\mathcal{M}' = \mathsf{aug}_{\delta_{\mathsf{c}}}(\mathcal{M}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{M}$ from $\langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. We can use the inductive assumption on $\mathcal{M}$ and conclude that either $\mathcal{M} = \mathfrak{X}$ or $\mathcal{M}$ belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. In the former case we immediately conclude that $\mathcal{M}' = \mathfrak{X}$. In the latter case, if $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, then $\mathcal{M}' = \mathfrak{X}$. Finally, if $\mathcal{M}$ does not belong to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, then

$$\mathcal{M}' = \mathsf{aug}_{\delta_{\mathsf{c}}}(\mathcal{M}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}) = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}) \enspace ,$$

so $\mathcal{M}'$ belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P}' \rangle\!\rangle_{\mathsf{RE}}$ because by assumption $\mathcal{M}$ belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. □

Lemma F.30. *Let $\boldsymbol{P}$ be a DLP, $\oplus_\beta$ a δ_{b}-based rule update operator and $\oplus_\gamma$ a δ_{c}-based rule update operator. If $\mathcal{M}$ belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$, then for some $\mathcal{N}$ from $\langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$, $\mathcal{N} \subseteq \mathcal{M}$.*

Proof. Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$. We will prove a stronger statement: If $\mathcal{M}$ belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$, then for some set of RE-rule-expressible sets of three-valued interpretations $\mathcal{S}$ and some $\mathcal{N} \in \langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$,

$$\mathcal{M} = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{N}, \mathcal{S}) \enspace .$$

We prove this by induction on n:

1° If $n = 0$, then the condition is vacuously satisfied.

2° Assuming that the statement holds for n, we prove it for $n+1$. Let $\boldsymbol{P}' = \langle \mathcal{P}_i \rangle_{i \leq n}$ and suppose that $\mathcal{M}'$ belongs to $\langle\!\langle \bigoplus_\beta \boldsymbol{P}' \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle (\bigoplus_\beta \boldsymbol{P}) \oplus_\beta \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, i.e. it belongs to

$$\left\{ \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}) \;\middle|\; \mathcal{M} \in \langle\!\langle \textstyle\bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \right\} \cup \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

If $\mathcal{M}'$ belongs to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, then it also belongs to $\langle\!\langle \bigoplus_\gamma \boldsymbol{P}' \rangle\!\rangle_{\mathsf{RE}}$ and our property follows from the fact that $\mathcal{M}' = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}', \emptyset)$.
In the principal case we know that $\mathcal{M}' = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{M}$ from $\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$. We can use the inductive assumption on $\mathcal{M}$ and conclude that there is some $\mathcal{N} \in \langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}}$ and some set of RE-rule-expressible sets of three-valued interpretations such that $\mathcal{M} = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{N}, \mathcal{S})$. We now consider two cases:

a) If $\mathcal{N}$ belongs to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, then $\mathcal{N}$ belongs to $\langle\!\langle \bigoplus_\gamma \boldsymbol{P}' \rangle\!\rangle_{\mathsf{RE}}$ and we are finished.

b) If $\mathcal{N}$ does not belong to $\langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}$, then

$$\mathsf{aug}_{\delta_{\mathsf{c}}}(\mathcal{N}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}) = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{N}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}) \enspace .$$

Let $\mathcal{N}' = \mathsf{aug}_{\delta_b}(\mathcal{N}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}})$. From the above we know that $\mathcal{N}'$ belongs to $\langle\!\langle \bigoplus_\gamma \boldsymbol{P}' \rangle\!\rangle_{\mathsf{RE}}$. Moreover, by Proposition F.24,

$$\begin{aligned}\mathcal{M}' &= \mathsf{aug}_{\delta_b}\left(\mathcal{M}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}\right) = \mathsf{aug}_{\delta_b}\left(\mathsf{aug}_{\delta_b}\left(\mathcal{N}, \mathcal{S}\right), \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}\right) \\ &= \mathsf{aug}_{\delta_b}\left(\mathcal{N}, \mathcal{S} \cup \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}\right) = \mathsf{aug}_{\delta_b}\left(\mathsf{aug}_{\delta_b}\left(\mathcal{N}, \langle\!\langle \mathcal{P}_n \rangle\!\rangle_{\mathsf{RE}}\right), \mathcal{S}\right) \\ &= \mathsf{aug}_{\delta_b}\left(\mathcal{N}', \mathcal{S}\right) \quad .\end{aligned}$$

□

Proposition F.31. *Let $\boldsymbol{P}$ be a DLP, $\oplus_\beta$ a δ_b-based rule update operator and $\oplus_\gamma$ a δ_c-based rule update operator. Then,*

$$\left[\!\left[\bigoplus_\beta \boldsymbol{P} \right]\!\right]_{\mathsf{SM}} = \left[\!\left[\bigoplus_\gamma \boldsymbol{P} \right]\!\right]_{\mathsf{SM}} \quad .$$

Proof. It follows from Lemma F.29 that

$$\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \cup \{ \mathfrak{X} \} \supseteq \langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \quad ,$$

so

$$[\![\bigoplus_\beta \boldsymbol{P}]\!]_{\mathsf{RE}} = \bigcap \langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} = \bigcap (\langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \cup \{ \mathfrak{X} \}) \subseteq \bigcap \langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} = [\![\bigoplus_\gamma \boldsymbol{P}]\!]_{\mathsf{RE}} \quad .$$

Furthermore, it follows from Lemma F.30 that

$$[\![\bigoplus_\beta \boldsymbol{P}]\!]_{\mathsf{RE}} = \bigcap \langle\!\langle \bigoplus_\beta \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} \supseteq \bigcap \langle\!\langle \bigoplus_\gamma \boldsymbol{P} \rangle\!\rangle_{\mathsf{RE}} = [\![\bigoplus_\gamma \boldsymbol{P}]\!]_{\mathsf{RE}} \quad .$$

Thus, $[\![\bigoplus_\beta \boldsymbol{P}]\!]_{\mathsf{RE}} = [\![\bigoplus_\gamma \boldsymbol{P}]\!]_{\mathsf{RE}}$. The rest follows from Proposition 7.30. □

Theorem 8.9. *Let $\oplus$ be a δ_b- or δ_c-based rule update operator. Then $\oplus$ respects support, fact update, causal rejection, acyclic justified update.*

Proof. This follows from Theorem 8.10 and from the fact that the justified update models satisfy all of these properties. □

Theorem 8.10. *Let $\boldsymbol{P}$ be a DLP, J an interpretation and $\oplus$ a δ_b- or δ_c-based rule update operator. Then,*

- $[\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{SM}} \subseteq [\![\boldsymbol{P}]\!]_{\mathsf{JU}}$ *and*
- *if* $\mathsf{all}(\boldsymbol{P})$ *contains no local cycles, then* $[\![\boldsymbol{P}]\!]_{\mathsf{JU}} \subseteq [\![\bigoplus \boldsymbol{P}]\!]_{\mathsf{SM}}$.

Proof. Follows from Propositions F.27, F.28 and F.31. □

Corollary 8.11. *Let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a DLP such that $\mathsf{all}(\boldsymbol{P})$ contains no local cycles, $\oplus$ be a δ_b- or δ_c-based rule update operator and $j < n$. Then there exists a rule base $\mathcal{R}$ such that $[\![\boldsymbol{P}]\!]_{\mathsf{JU}} = [\![\bigoplus \boldsymbol{P}']\!]_{\mathsf{SM}}$ where $\boldsymbol{P}' = \langle \mathcal{R}, \mathcal{P}_{j+1}, \ldots, \mathcal{P}_{n-1} \rangle$.*

Proof. It suffices to put $\mathcal{R} = \bigoplus \langle \mathcal{P}_i \rangle_{i \leq j}$ and apply Theorem 8.10. □

F.1.5 Semantic Properties

Proposition F.32. *Let $\oplus$ be a δ-based rule update operator. Then $\oplus$ satisfies* (Initialisation), (Disjointness), (P1) *and* (P2.⊤) *with respect to* RR, SR, RMR, SMR, RE, SE *and* SM *(where applicable).*

Proof. We prove these properties with respect to RR-equivalence; their satisfaction with respect to the other notions of program equivalence follows from Proposition 7.30.

To verify that (Initialisation) holds, it suffices to observe that

$$\langle\!\langle \emptyset \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \{\, \mathsf{aug}_{\delta}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \emptyset \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

Thus, $\emptyset \oplus \mathcal{U}$ is RR-equivalent to $\mathcal{U}$.

Turning to (Disjointness), it suffices to observe that $\langle\!\langle (\mathcal{R} \cup \mathcal{S}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ coincides with

$$\begin{aligned} &\{\, \mathsf{aug}_{\delta} (\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \mathcal{R} \cup \mathcal{S} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \\ &\quad = (\{\, \mathsf{aug}_{\delta} (\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \mathcal{R} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \cup (\{\, \mathsf{aug}_{\delta} (\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \mathcal{S} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \\ &\quad = \langle\!\langle \mathcal{R} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \cup \langle\!\langle \mathcal{S} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup (\mathcal{S} \oplus \mathcal{U}) \rangle\!\rangle_{\mathsf{RE}} \enspace . \end{aligned}$$

Note that we did not need to use the assumption that $\mathcal{R}$, $\mathcal{S}$ are over disjoint alphabets.

In order to prove that (P1) holds, consider that $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ is a subset of $\langle\!\langle \mathcal{R} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. Consequently, $\mathcal{R} \oplus \mathcal{U}$ RR-entails $\mathcal{U}$.

Finally, (P2.⊤) follows from the fact that $\mathsf{aug}_{\delta}(\mathcal{M}, \emptyset) = \mathcal{M}$ for all $\mathcal{M} \subseteq \mathfrak{X}$. □

Lemma F.33. *Let $\mathcal{P}$, $\mathcal{Q}$ be programs over disjoint alphabets. Then for all $\mathcal{M} \in \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}$,*

$$\mathsf{aug}_{\delta_{\mathsf{a}}}(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M} \enspace .$$

Also, for all non-empty sets $\mathcal{M} \in \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}$,

$$\mathsf{aug}_{\delta_{\mathsf{c}}}(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$$

and

$$\mathsf{aug}_{\delta_{\mathsf{c}}}(\emptyset, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \begin{cases} \mathfrak{X} & \emptyset \in \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}} \enspace ; \\ \emptyset & \textit{otherwise} \enspace . \end{cases}$$

Proof. Take some $\pi \in \mathcal{P}$ and some $\sigma \in \mathcal{Q}$ and put $\mathcal{M} = [\![\pi]\!]_{\mathsf{RE}}$, $\mathcal{N} = [\![\sigma]\!]_{\mathsf{RE}}$. We first show that for all atoms p and interpretations J, $\mathcal{M} \bowtie^{p}_{J} \mathcal{N}$ does not hold. By definition this holds if and only if $\mathcal{M}^{J}(p) \neq \mathcal{N}^{J}(p)$. But according to Proposition F.9, this is possible only if p occurs in the heads of both π and σ, contrary to our assumption.

Consequently, by the definitions of δ_{a} and δ_{b},

$$\delta_{\mathsf{a}}(\mathcal{M}, \mathcal{N}) = \delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N}) = \emptyset \enspace ,$$

so

$$\mathsf{aug}_{\delta_{\mathsf{a}}}(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M} \enspace .$$

Furthermore, $\delta_{\mathsf{c}}(\mathcal{M}, \mathcal{N})$ differs from $\delta_{\mathsf{b}}(\mathcal{M}, \mathcal{N})$ if and only if $\mathcal{M} = \mathcal{N}$. In this case, π is RE-equivalent to σ, so $\mathsf{can}_{\mathsf{RE}}(\pi) = \mathsf{can}_{\mathsf{RE}}(\sigma)$. Also, since π and σ are over different alphabets, it follows from the definition of $\mathsf{can}_{\mathsf{RE}}(\cdot)$ that this is possible if and only if either both π and σ are tautological, or if $H(\pi) = B(\pi) = H(\sigma) = B(\sigma) = \emptyset$. In the former case we can observe that

$$\mathsf{aug}_{\delta_{\mathsf{c}}}(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathfrak{X}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathfrak{X} = \mathcal{M} \enspace .$$

In the latter case we have $\mathcal{M} = \mathcal{N} = \emptyset$, and

$$\mathsf{aug}_{\delta_{\mathsf{c}}}(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathfrak{X} \enspace .$$

□

Corollary F.34. *Let $\mathcal{P}$, $\mathcal{Q}$ be programs over disjoint alphabets and δ be either δ_a, δ_b or δ_c. Then for all $\mathcal{M} \in \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}$, either $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$, or $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{Q} \rangle\!\rangle_{\mathsf{RE}}) = \mathfrak{X}$.*

Proof. Follows directly from Lemma F.33. □

Lemma F.35. *Let $\mathcal{M}$ be an RE-rule-expressible set of three-valued interpretations, $\mathcal{U}$ a program, p an atom and V_0 a truth value.*

If $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})^J(p) = \mathsf{V}_0$, then either $\mathcal{M}^J(p) = \mathsf{V}_0$, or p occurs in $\mathcal{U}$.

Proof. Suppose that $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})^J(p) = \mathsf{V}_0$ and $\mathcal{M}^J(p) \neq \mathsf{V}_0$. Thus, $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ contains $J[p/\mathsf{V}]$ if and only if $\mathsf{V} = \mathsf{V}_0$. Since $\mathcal{M}$ is a subset of $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$, it follows that $\mathcal{M}$ contains neither $J[\mathsf{T}/p]$ nor $J[\mathsf{U}/p]$ nor $J[\mathsf{F}/p]$. Let π be some rule such that $[\![\pi]\!]_{\mathsf{RE}} = \mathcal{M}$. We can conclude that p does not occur in π. Furthermore, if $\mathsf{V}_0 = \mathsf{T}$, then by the definition of δ_a we obtain a conflict with the fact that $J[\mathsf{U}/p]$ does not belong to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$. Consequently, $\mathsf{V}_0 = \mathsf{F}$. Furthermore, since $J[\mathsf{T}/p]$ does not belong to $\mathsf{aug}_{\delta_\mathsf{a}}(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ but $J[\mathsf{F}/p]$ does, there exists some $\mathcal{N} \in \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ and some atom q such that $\mathcal{M} \bowtie_q^{J[p/\mathsf{F}]} \mathcal{N}$ but it is not the case that $\mathcal{M} \bowtie_q^{J[p/\mathsf{T}]} \mathcal{N}$. Since p does not occur in π, this is only possible if p occurs in $\mathcal{U}$. □

Proposition F.36. *Let $\oplus$ be a δ_a-, δ_b- or δ_c-based rule update operator. Then $\oplus$ satisfies (Non-interference) for non-disjunctive programs with respect to RR, SR, RMR, SMR, RE, SE and SM.*

Proof. We prove this property with respect to RR-equivalence; its satisfaction with respect to the other notions of program equivalence follows from Proposition 7.30.

Suppose that $\mathcal{P}, \mathcal{U}$ and $\mathcal{V}$ are non-disjunctive programs, δ is either δ_a, δ_b or δ_c and $\oplus$ is a δ-based rule update operator. Take some set

$$\mathcal{M}_0 \in \langle\!\langle (\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

We will show that $\mathcal{M}_0$ either belongs to $\langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ or $\mathcal{M}_0 = \mathfrak{X}$. Note that $\langle\!\langle (\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$ coincides with

$$\{\, \mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}\right) \mid \mathcal{M} \in \langle\!\langle \mathcal{P} \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \,\} \cup \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}} \enspace .$$

We need to consider three cases. The first one occurs when $\mathcal{M}_0$ belongs to $\langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$. Then $\mathcal{M}_0$ also belongs to $\langle\!\langle \mathcal{P} \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}$ and since $\langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$ coincides with

$$\{\, \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) \mid \mathcal{M} \in \langle\!\langle \mathcal{P} \oplus \mathcal{V} \rangle\!\rangle_{\mathsf{RE}} \,\} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \enspace ,$$

it must contain the set $\mathsf{aug}_\delta(\mathcal{M}_0, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$. Furthermore, since $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets, it follows Corollary F.34 that $\mathsf{aug}_\delta(\mathcal{M}_0, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ is either $\mathcal{M}_0$ or $\mathfrak{X}$. In either case, our proof ends.

The second case occurs when $\mathcal{M}_0 = \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}})$ and $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$. As in the previous case, since $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets, it follows from Corollary F.34 that $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}})$ is either $\mathcal{M}$ or $\mathfrak{X}$. In the former case, $\mathcal{M}_0$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$, so it certainly belongs to $\langle\!\langle (\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}$.

The last case occurs when

$$\mathcal{M}_0 = \mathsf{aug}_\delta\left(\mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle \mathcal{V} \rangle\!\rangle_{\mathsf{RE}}\right)$$

for some $\mathcal{M} \in \langle\!\langle \mathcal{P} \rangle\!\rangle_{\mathsf{RE}}$.

If δ is δ_b, then it follows directly from Lemma F.23 that

$$\begin{aligned}\mathcal{M}_0 &= \mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}} \cup \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right) \\ &= \mathsf{aug}_\delta\left(\mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right) \quad .\end{aligned}$$

Consequently, $\mathcal{M}_0$ belongs to $\langle\!\langle(\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$.

If δ is δ_c, then either $\mathcal{M}_0 = \mathfrak{X}$ or

$$\mathcal{M}_0 = \mathsf{aug}_{\delta_b}\left(\mathsf{aug}_{\delta_b}\left(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right) \quad .$$

The rest follows by the previous paragraph.

If δ is δ_a, then we consider two subcases:

a) If $\mathsf{aug}_{\delta_a}(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$, then $\mathcal{M}_0 = \mathsf{aug}_{\delta_a}(\mathcal{M}, \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}})$, so $\mathcal{M}_0$ belongs to $\langle\!\langle\mathcal{P} \oplus \mathcal{V}\rangle\!\rangle_{\mathsf{RE}}$. Take some $\mathcal{N} \in \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$ and suppose that $\mathcal{M}_0 \bowtie^p_J \mathcal{N}$. Then, by Lemma F.35, for some truth value V_0 either $\mathcal{M}^J(p) = \mathsf{V}_0$, or p occurs in $\mathcal{V}$. In the former case we obtain a conflict with the assumption that $\mathsf{aug}_{\delta_a}(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$ while the latter case is in conflict with the assumption that $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets. Thus, no such $\mathcal{N}$ exists and $\mathsf{aug}_{\delta_a}(\mathcal{M}_0, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}_0$. Consequently, $\mathcal{M}_0$ belongs to $\langle\!\langle(\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$.

b) If $\mathsf{aug}_{\delta_a}(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}) \neq \mathcal{M}$, then there is some $\mathcal{N} \in \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$ such that $\mathcal{M} \bowtie^p_J \mathcal{N}$ for some atom p and interpretation J. Thus since $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets, it follows from Proposition F.10 that $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$, so $\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}})$ belongs to $\langle\!\langle(\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U}\rangle\!\rangle_{\mathsf{RE}}$.
It remains to show that $\mathcal{M}_0 = \mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}})$. Put $\mathcal{M}_1 = \mathsf{aug}_{\delta_a}(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}})$ and suppose that for some $\mathcal{N}' \in \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}$, some atom q and some interpretation K, $\mathcal{M}_1 \bowtie^q_K \mathcal{N}'$. Then, by Lemma F.35, either $\mathcal{M}^K(q) = \mathsf{V}_0$ for some truth value V_0, or q occurs in $\mathcal{U}$. In the former case, $p = q$ by Proposition F.10 and we obtain a conflict with the fact that $\mathcal{U}$ and $\mathcal{V}$ are over disjoint alphabets. In the latter case, q occurs in both $\mathcal{U}$ and $\mathcal{V}$, so the same conflict follows. Consequently, there is no such $\mathcal{N}'$ and we can conclude that

$$\begin{aligned}\mathcal{M}_0 &= \mathsf{aug}_\delta\left(\mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right) \\ &= \mathsf{aug}_\delta\left(\mathcal{M}_1, \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right) \\ &= \mathsf{aug}_\delta\left(\mathcal{M}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right) \quad .\end{aligned}$$

The proof of the other inclusion is symmetric. □

Example F.37 ((Non-interference) for Disjunctive Programs)**.** Consider the programs

$$\mathcal{P}: \quad p; q; r. \qquad\qquad \mathcal{U}: \quad \begin{aligned} \sim p &\leftarrow \sim r. \\ p &\leftarrow r. \\ r &\leftarrow p. \end{aligned} \qquad\qquad \mathcal{V}: \quad \sim q.$$

Let the rule in $\mathcal{P}$ be denoted by π. By following the definition of δ_b, we can conclude that

$$\begin{aligned}\mathsf{aug}_{\delta_b}\left(\mathsf{aug}_{\delta_b}\left([\![\pi]\!]_{\mathsf{RE}}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right) &= \mathsf{aug}_{\delta_b}\left([\![\pi]\!]_{\mathsf{RE}}, \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right) \\ &= [\![\pi]\!]_{\mathsf{RE}} \cup \{\, (\emptyset, \emptyset), (\emptyset, p), (\emptyset, q), (\emptyset, r), (\emptyset, qr) \,\} \quad , \\ \mathsf{aug}_{\delta_b}\left(\mathsf{aug}_{\delta_b}\left(\pi, \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right), \langle\!\langle\mathcal{U}\rangle\!\rangle_{\mathsf{RE}}\right) &= \mathsf{aug}_{\delta_b}\left([\![\pi]\!]_{\mathsf{RE}}, \langle\!\langle\mathcal{V}\rangle\!\rangle_{\mathsf{RE}}\right) \\ &= [\![\pi]\!]_{\mathsf{RE}} \cup \{\, (\emptyset, \emptyset), (\emptyset, p), (\emptyset, q), (\emptyset, r), (\emptyset, pr) \,\} \quad .\end{aligned}$$

Thus, $(\emptyset, pr)$ does not belong to $[\![(\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V}]\!]_{\mathsf{RE}}$ although it belongs to $[\![(\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U}]\!]_{\mathsf{RE}}$. Also, (pr, pr) is an RE-model of all rules in $\mathcal{P}$, $\mathcal{U}$ and $\mathcal{V}$ and both (p, p, r) and (r, pr) are not RE-models of $\mathcal{U}$. The consequence is that although $\{ p, r \}$ is a stable model of $(\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V}$, it is not a stable model of $(\mathcal{P} \oplus \mathcal{V}) \oplus \mathcal{U}$. The results are the same when δ_{c} is used instead of δ_{b}. So δ_{b}-based and δ_{c}-based rule update operators do not satisfy (Non-interference) for disjunctive programs w.r.t. SM-equivalence nor any stronger notion of program equivalence.

Turning to δ_{a}, the above example does not work as it works for δ_{b} and δ_{c}. But consider these rules instead:

$$\pi_0 = (p; q.) \ , \qquad \pi_1 = (\sim p \leftarrow \sim r.) \ , \qquad \pi_2 = (\sim q \leftarrow s.) \ .$$

Let $\mathcal{P}' = \{ \pi_0 \}, \mathcal{U}' = \{ \pi_1 \}$ and $\mathcal{V}' = \{ \pi_2 \}$. It is not hard to verify that

$$\begin{aligned} &\mathsf{aug}_\delta \left([\![\pi_0]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{U}' \rangle\!\rangle_{\mathsf{RE}} \right) = [\![\pi_0]\!]_{\mathsf{RE}} \cup \{ (I, J) \mid J \subseteq \{ p, s \} \} \ , \\ &\mathsf{aug}_\delta \left(\mathsf{aug}_\delta \left([\![\pi_0]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{U}' \rangle\!\rangle_{\mathsf{RE}} \right), \langle\!\langle \mathcal{V}' \rangle\!\rangle_{\mathsf{RE}} \right) \\ &\qquad = [\![\pi_0]\!]_{\mathsf{RE}} \cup \{ (I, J) \mid J \subseteq \{ p, s \} \} \cup \{ (I, J) \mid \{ s \} \subseteq J \subseteq \{ q, r, s \} \} \end{aligned}$$

while

$$\begin{aligned} &\mathsf{aug}_\delta \left([\![\pi_0]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{V}' \rangle\!\rangle_{\mathsf{RE}} \right) = [\![\pi_0]\!]_{\mathsf{RE}} \cup \{ (I, J) \mid \{ s \} \subseteq J \subseteq \{ q, r, s \} \} \ , \\ &\mathsf{aug}_\delta \left(\mathsf{aug}_\delta \left([\![\pi_0]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{V}' \rangle\!\rangle_{\mathsf{RE}} \right), \langle\!\langle \mathcal{U}' \rangle\!\rangle_{\mathsf{RE}} \right) \\ &\qquad = [\![\pi_0]\!]_{\mathsf{RE}} \cup \{ (I, J) \mid \{ s \} \subseteq J \subseteq \{ q, r, s \} \} \cup \{ (I, J) \mid J \subseteq \{ p \} \} \ . \end{aligned}$$

As a consequence, for any δ_{a}-based rule update operator $\oplus$ it holds that while $(\emptyset, ps)$ belongs to $\langle\!\langle (\mathcal{P}' \oplus \mathcal{U}') \oplus \mathcal{V}' \rangle\!\rangle_{\mathsf{RE}}$, it does not belong to $\langle\!\langle (\mathcal{P}' \oplus \mathcal{V}') \oplus \mathcal{U}' \rangle\!\rangle_{\mathsf{RE}}$. This already proves that (Non-interference) does not hold for δ_{a}-based operators for disjunctive programs w.r.t. RE-equivalence. This counterexample equally applies to SE-equivalence because the set of three-valued interpretations

$$\mathsf{aug}_\delta \left(\mathsf{aug}_\delta \left([\![\pi_0]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{U}' \rangle\!\rangle_{\mathsf{RE}} \right), \langle\!\langle \mathcal{V}' \rangle\!\rangle_{\mathsf{RE}} \right)$$

is RE-expressible by the program

$$\begin{aligned} p; q; \sim r &\leftarrow \sim s. \\ p; q; \sim p; \sim q&. \end{aligned}$$

while the set

$$\mathsf{aug}_\delta \left(\mathsf{aug}_\delta \left([\![\pi_0]\!]_{\mathsf{RE}}, \langle\!\langle \mathcal{V}' \rangle\!\rangle_{\mathsf{RE}} \right), \langle\!\langle \mathcal{U}' \rangle\!\rangle_{\mathsf{RE}} \right)$$

is RE-expressible by the program

$$\begin{aligned} p; q; \sim r &\leftarrow \sim s. \\ p; q; \sim p; \sim q&. \\ p; q; \sim p; \sim s &\leftarrow \sim r. \end{aligned}$$

and $(\emptyset, ps)$ is an SE-model of the former program but not of the latter (as it is of the rules in $\mathcal{U}'$ and $\mathcal{V}'$). Thus, (Non-interference) does not hold for δ_{a}-based operators for disjunctive programs w.r.t. SE-equivalence or any stronger notion of program equivalence.

However, this still does not prove that it does not hold w.r.t. SM-equivalence. This is because (ps, ps) is not an RE-model of π_1. It so seems that this cannot be circumvented and it is probably possible to prove that δ_a-based operators satisfy (Non-interference) w.r.t. SM-equivalence. A proof of this would probably need to start by showing that if $(J, J) \notin \mathcal{M}$ and $(J, J) \in \mathsf{aug}_{\delta_a}(\mathcal{M}, \mathcal{S})$, then $(J, J) \in \mathsf{aug}_{\delta_a}(\mathsf{aug}_{\delta_a}(\mathcal{M}, \mathcal{T}), \mathcal{S})$. This, intuitively, proves that by performing an update by $\mathcal{T}$ before $\mathcal{S}$, one can only "avoid" introducing interpretations (I, J) for which (J, J) does not belong to $\bigcap \mathcal{S}$ (because of the way conflicts are identified). These differences are then not detectable on the level of stable models. The correctness of this approach to the proof needs some further work. However, interestingly, the differences between $(\mathcal{P}' \oplus \mathcal{U}') \oplus \mathcal{V}'$ and $(\mathcal{P}' \oplus \mathcal{V}') \oplus \mathcal{U}'$ can be "revealed" by further updates. For instance, an update by $\{ p \leftarrow s., s \leftarrow p. \}$ weakens the rule π_1 and in the former case $\{ p, s \}$ becomes a stable model after this update while in the latter case it does not.

Proposition F.38. *Let $\oplus$ be a δ-based rule update operator where $\delta(\mathcal{M}, \mathfrak{X}) \subseteq \mathcal{M}$ for all $\mathcal{M} \subseteq \mathfrak{X}$. Then $\oplus$ satisfies (Tautology) and (Immunity to Tautologies) with respect to RR, SR, RMR, SMR, RE, SE and SM.*

Proof. For RR-equivalence this can be verified in a straight-forward manner. For the remaining notions of program equivalence this follows from Proposition 7.30. □

Proposition F.39. *Let $\oplus$ be a δ-based rule update operator. Then $\oplus$ satisfies (Idempotence) with respect to RMR, SMR, RE, SE and SM.*

Moreover, if $\oplus$ is δ_c-based, then it also satisfies (Idempotence) with respect to RR and SR.

Proof. (Idempotence) states the following: $\mathcal{R} \oplus \mathcal{R} \equiv \mathcal{R}$. We will show that this is true under RMR-equivalence which, together with Proposition 7.30, implies that it holds under SMR-, RE-, SE- and SM-equivalence.

So first take some $\mathcal{M} \in \min\langle\!\langle (\mathcal{R} \oplus \mathcal{R}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. If $\mathcal{M} = \mathfrak{X}$, then $\mathcal{R} \oplus \mathcal{R}$ is tautological and since $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ is a subset of $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, $\mathcal{R}$ is itself tautological. Thus, $\mathfrak{X}$ also belongs to $\min\langle\!\langle \mathcal{R} \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. In the principal case either $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, or it is equal to $\mathsf{aug}_{\delta}(\mathcal{M}_0, \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{M}_0 \in \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$. In the latter case we have that $\mathcal{M}_0$ is a subset of $\mathcal{M}$ and since $\mathcal{M}_0$ belongs to $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, by the minimality of $\mathcal{M}$ we obtain that $\mathcal{M} = \mathcal{M}_0$, so $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$. Now it follows that $\mathcal{M}$ is minimal $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ because $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ is a subset of $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ and $\mathcal{M}$ is minimal in the latter set.

Now take some $\mathcal{M} \in \min\langle\!\langle \mathcal{R} \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. If $\mathcal{M} = \mathfrak{X}$, then $\mathcal{R}$ is tautological and it follows by the properties of $\oplus$ that $\mathcal{R} \oplus \mathcal{R}$ is also tautological. Thus, $\mathfrak{X}$ also belongs to $\langle\!\langle (\mathcal{R} \oplus \mathcal{R}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. In the principal case $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$. Take some $\mathcal{N} \in \langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$ such that $\mathcal{N}$ is a subset of $\mathcal{M}$. If $\mathcal{N}$ belongs to $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, then it follows by minimality of $\mathcal{M}$ that $\mathcal{M} = \mathcal{N}$. On the other hand, if $\mathcal{N}$ is of the form $\mathsf{aug}_{\delta}(\mathcal{N}_0, \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{N}_0$ from $\langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$, then

$$\mathcal{N}_0 \subseteq \mathcal{N} \subseteq \mathcal{M} \enspace ,$$

so by the minimality of $\mathcal{M}$, $\mathcal{N}_0 = \mathcal{N} = \mathcal{M}$. Thus, in either case $\mathcal{M}$ is equal to $\mathcal{N}$, which proves that it is minimal within $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}}$.

Now consider some δ_c-based rule update operator $\oplus$. Then $\langle\!\langle \mathcal{R} \oplus \mathcal{R} \rangle\!\rangle_{\mathsf{RE}} \subseteq \langle\!\langle \mathcal{R} \rangle\!\rangle_{\mathsf{RE}} \cup \{ \mathfrak{X} \}$, so obviously $\langle\!\langle (\mathcal{R} \oplus \mathcal{R}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle \mathcal{R} \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. □

A counterexample showing that (Idempotence) does not hold for δ_a- and δ_b-based rule update operators follows:

Example F.40 ((Idempotence) under $\equiv_{\mathsf{SR}}$ and $\equiv_{\mathsf{RR}}$)**.** Let $\pi = (p \leftarrow {\sim}q.)$ and $\sigma = ({\sim}p \leftarrow r.)$. The potential conflict between π and σ is detected the same way by both δ_a and δ_b: when

π is updated by σ, it becomes weakened. In particular, the weakened version of π is $\pi' = (p \leftarrow \sim q, \sim r.)$ (or another RE-equivalent rule) because

$$\llbracket \pi' \rrbracket_{\mathsf{RE}} = \llbracket \pi \rrbracket_{\mathsf{RE}} \cup \delta_{\mathsf{a}}(\llbracket \pi \rrbracket_{\mathsf{RE}}, \llbracket \sigma \rrbracket_{\mathsf{RE}}) \\ = \llbracket \pi \rrbracket_{\mathsf{RE}} \cup \delta_{\mathsf{b}}(\llbracket \pi \rrbracket_{\mathsf{RE}}, \llbracket \sigma \rrbracket_{\mathsf{RE}}) \enspace .$$

Note that (Idempotence) states the following: $\mathcal{R} \oplus \mathcal{R} \equiv \mathcal{R}$. Let's put $\mathcal{R} = \{ \pi, \sigma \}$ and consider some δ_{a}- or δ_{b}-based update operator $\oplus$. What is the result of $\mathcal{R} \oplus \mathcal{R}$? It will certainly contain both rules from $\mathcal{R}$ because $\mathcal{R}$ is the updating rule base. But it will also contain weakened versions of π and σ because these two rules are in a potential conflict. In particular, it will contain π' (or another RE-equivalent rule). This implies that under SR- and RR-equivalence $\mathcal{R} \oplus \mathcal{R}$ is not considered equivalent to $\mathcal{R}$. This is because π' is neither SE- nor RE-equivalent to some rule in $\mathcal{R}$, nor is it tautological. Thus, (Idempotence) does not hold for δ_{a}- and δ_{b}-based update operators under $\equiv_{\mathsf{SR}}$ and $\equiv_{\mathsf{RR}}$.

This issue is resolved by δ_{c} by weakening π all the way to a tautological rule. Thus, when $\oplus$ is δ_{c}-based, $\mathcal{R} \oplus \mathcal{R}$ contains instead of π' a tautological rule which then gets ignored by SR- and RR-equivalence.

Lemma F.41. *Let $\mathcal{M}$ be an RE-rule-expressible set of three-valued interpretations, $\mathcal{S}$ a set of RE-rule-expressible sets of three-valued interpretations and δ be either δ_{b} or δ_{c}. Then,*

$$\mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{S}) \in \{ \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{X} \} \enspace .$$

Proof. Put $\mathcal{M}' = \mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$ and take some $\mathcal{N} \in \mathcal{S}$ such that $\mathcal{M}' \bowtie_J^p \mathcal{N}$ for some atom p and some interpretation J. Then $\mathcal{M}'^J(p) = \mathsf{V}_0$ for some truth value V_0, so it follows from Lemma F.23 that $\mathcal{M}^J(p) = \mathsf{V}_0$. But this implies $\mathcal{M} \bowtie_J^p \mathcal{N}$, so both $(J \setminus \{ p \}, J \setminus \{ p \})$ and $(J \cup \{ p \}, J \cup \{ p \})$ belong to $\mathcal{M}'$, a conflict with the assumption that $\mathcal{M}'^J(p)$ is defined. As a consequence, no such $\mathcal{N} \in \mathcal{S}$ exists, so $\mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}', \mathcal{S}) = \mathcal{M}' = \mathsf{aug}_{\delta_{\mathsf{b}}}(\mathcal{M}, \mathcal{S})$.

Turning to δ_{c}, we can observe that either the previous case applies, or

$$\mathsf{aug}_\delta(\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S}), \mathcal{S}) = \mathcal{X} \enspace .$$

□

Proposition F.42. *Let $\oplus$ be a δ_{b}- or δ_{c}-based rule update operator. Then $\oplus$ satisfies (Absorption) with respect to RMR, SMR, RE, SE and SM.*

Moreover, if $\oplus$ is δ_{c}-based, then it also satisfies (Absorption) with respect to RR and SR.

Proof. (Absorption) states the following: $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U} \equiv \mathcal{R} \oplus \mathcal{U}$. We will show that this is true under RMR-equivalence which, together with Proposition 7.30, implies that it holds under SMR-, RE-, SE- and SM-equivalence.

So suppose that $\oplus$ is δ_{b}- or δ_{c}-based. It follows directly from Lemma F.41 that $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$ is a superset of $\langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. Thus, whenever $\mathcal{M}$ is minimal in $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$, it must also be minimal in $\langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. Furthermore, the extra elements of $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$ are never smaller than the elements of $\langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$ because they are of the form $\mathsf{aug}_\delta(\mathcal{N}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}})$ for some $\mathcal{N} \in \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}} \subseteq \langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. Thus, whenever $\mathcal{M}$ is minimal in $\langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$, it must also be minimal in $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$.

Furthermore, if $\oplus$ is δ_{c}-based, then it follows from the above that $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}} = \langle\!\langle (\mathcal{R} \oplus \mathcal{U}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$. □

Proposition F.43. *Let $\oplus$ be a δ_{b}- or δ_{c}-based rule update operator. Then $\oplus$ satisfies (Augmentation) for non-disjunctive programs with respect to RMR, SMR, RE, SE and SM.*

Moreover, if $\oplus$ is δ_{c}-based, then it also satisfies (Augmentation) for non-disjunctive programs with respect to RR and SR.

Proof. The proof for RMR-equivalence follows from Proposition F.24 and Lemma F.29 and from the the fact that the extra elements of $\langle\!\langle ((\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$, as compared to $\langle\!\langle (\mathcal{R} \oplus \mathcal{V}) \cup \{ \tau \} \rangle\!\rangle_{\mathsf{RE}}$ are non-minimal in the latter set.

If $\oplus$ is δ_{c}-based, then there are no extra elements and the rest follows from Proposition F.24 and Lemma F.29. □

Example F.44 ((Absorption) and (Augmentation) violated by δ_{a})**.** Consider the same rules as in Example 8.7, i.e.

$$\pi_0 = (p.) \ , \qquad \pi_1 = (\sim p \leftarrow \sim q.) \ , \qquad \pi_2 = (q.) \ .$$

Put $\mathcal{P} = \{ \pi_0 \}$ and $\mathcal{U} = \{ \pi_1, \pi_2 \}$ and take some δ_{a}-based rule update operator $\oplus$. Given the considerations of Example 8.7, it is not difficult to verify that $\mathcal{P} \oplus \mathcal{U}$ has the unique stable model $\{ p, q \}$ while $(\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{U}$ admits the stable model $\{ q \}$. As a consequence, $\oplus$ cannot satisfy (Absorption) nor (Augmentation) w.r.t. SM-equivalence or any stronger notion of equivalence.

Example F.45 ((Augmentation) for Disjunctive Programs)**.** Consider the programs:

$$\mathcal{P}: \quad p; q; r. \qquad \mathcal{U}: \quad \begin{array}{c} \sim p \leftarrow \sim r. \\ p \leftarrow r. \\ r \leftarrow p. \end{array} \qquad \mathcal{V}: \quad \begin{array}{c} \sim p \leftarrow \sim r. \\ p \leftarrow r. \\ r \leftarrow p. \\ \sim q. \end{array}$$

By reasoning very similar to the one in Example F.37 we can conclude that while $(\emptyset, pr)$ does not belong to $[\![(\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V}]\!]_{\mathsf{RE}}$, it belongs to $[\![\mathcal{P} \oplus \mathcal{V}]\!]_{\mathsf{RE}}$. The consequence is that while $\{ p, r \}$ is a stable model of $(\mathcal{P} \oplus \mathcal{U}) \oplus \mathcal{V}$, it is not a stable model of $\mathcal{P} \oplus \mathcal{V}$. Thus, (Augmentation) does not hold for $\oplus$ under SM-equivalence nor any stronger notion of program equivalence.

Example F.46 ((Associativity) and δ_{a}, δ_{b}, δ_{c})**.** The rule $\pi = (\sim p.)$, when updated by $\sigma = (p \leftarrow q.)$, must be weakened, anticipating the potential conflict. In the case of δ_{a}-, δ_{b}- and δ_{c}-based operators, the resulting rule is $\pi' = (\sim p \leftarrow \sim q.)$ (or another RE-equivalent rule). Consider the following rule bases:

$$\mathcal{R} = \{ p. \} , \quad \mathcal{U} = \{ \sim p. \} , \quad \mathcal{V} = \{ p \leftarrow q., q \leftarrow p. \} \ .$$

Note that (Associativity) states the following: $\mathcal{R} \oplus (\mathcal{U} \oplus \mathcal{V}) \equiv (\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V}$. However, while in $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V}$ the fact from $\mathcal{R}$ is completely annihilated (i.e. transformed into a tautological rule) due to the negative fact π in $\mathcal{U}$, this does not happen in $\mathcal{R} \oplus (\mathcal{U} \oplus \mathcal{V})$ because the π is first weakened into π'. As a consequence, $\mathcal{R} \oplus (\mathcal{U} \oplus \mathcal{V})$ has one extra stable model comparing to $(\mathcal{R} \oplus \mathcal{U}) \oplus \mathcal{V}$: $\{ p, q \}$. This implies that (Associativity) does not hold for δ_{a}-, δ_{b}- and δ_{c}-based rule update operators under SM-equivalence, nor under any stronger equivalence.

Proposition F.47. *Let $\oplus$ be a δ-based rule update operator. Then $\oplus$ satisfies (P2.1) and (P5) with respect to RMR, SMR, RE and SE.*

Proof. Under RMR-entailment (P2.1) follows from the fact that $\mathcal{M}$ is a subset of $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$ and (P5) follows from the fact that $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S})$ is a subset of $\mathsf{aug}_\delta(\mathcal{M}, \mathcal{S} \cup \mathcal{T})$. For the remaining notions of program entailment this follows from Proposition 7.30. □

A counterexample under RR- and SR-equivalence follows:

Example F.48 ((P2.1) under $\models_{\mathsf{SR}}$ and $\models_{\mathsf{RR}}$)**.** Consider again the rules π, σ from Example F.46 and rule bases $\mathcal{R} = \{ \pi \}, \mathcal{U} = \{ \sigma \}$. Note that (P2.1) states the following: $\mathcal{R} \cup \mathcal{U} \models \mathcal{R} \oplus \mathcal{U}$. However, if $\oplus$ is δ_{a}-, δ_{b}- or δ_{c}-based, $\mathcal{R} \oplus \mathcal{U}$ will contain π' (or another RE-equivalent rule) which results from weakening of π by σ. Consequently, when SR- or RR-entailment is used, $\mathcal{R} \cup \mathcal{U}$ cannot entail $\mathcal{R} \oplus \mathcal{U}$ simply because π' (or another equivalent rule) does not belong to $\mathcal{R} \cup \mathcal{U}$.

Example F.49 ((P2.2) and Rule Update Semantics)**.** Consider $\mathcal{R} = \{ p. \}$ and $\mathcal{U} = \{ {\sim}p. \}$ and note that (P2.2) states the following: $(\mathcal{R} \cup \mathcal{U}) \oplus \mathcal{U} \models \mathcal{R}$. In other words, it requires that

$$\{ p., {\sim}p. \} \oplus \{ {\sim}p. \} \models p \enspace .$$

In the presence of (P1) this amounts to postulating that one can never recover from an inconsistent state. Such a requirement is out of line with the way these situations are treated in state-of-the-art approaches to rule update which allow for recovery from an inconsistent state if all involved conflicts are resolved by the update. Note that, though for different reasons, (P2.2) has also been subject of harsh criticism in belief update literature (Herzig and Rifi, 1999).

Proposition F.50. *Let $\oplus$ be a δ-based rule update operator where $\delta(\mathcal{M}, \mathcal{X}) \subseteq \mathcal{M}$ for all $\mathcal{M} \subseteq \mathcal{X}$. Then $\oplus$ satisfies (P4), (P4.1), (P4.2) and (P8.2) with respect to RR.*

Proof. Principles (P4.2) and (P8.2) can be verified straightforwardly and (P4.1) as well as (P4) are their consequences. The condition on δ is necessary to ensure that

$$\mathsf{aug}_\delta(\mathcal{M}, \langle\!\langle \mathcal{U} \rangle\!\rangle_{\mathsf{RE}}) = \mathcal{M}$$

whenever $\mathcal{U}$ is tautological. □

For a discussion of these principles under weaker notions of program equivalence see the end of Section 8.1.

F.2 Belief Updates Using Exception-Based Operators

F.2.1 Model-Based Update Operators

Theorem 8.14. *If $\diamond$ is an update operator that satisfies (FO1), (FO2.1) and (FO4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of theories $\boldsymbol{T}$, $[\![\diamond \boldsymbol{T}]\!] = [\![\bigoplus \boldsymbol{T}]\!]$.*

Proof. Let the exception function ε be defined for all sets of interpretations $\mathcal{M} \subseteq \mathfrak{I}$ and all sets of sets of interpretations $\mathcal{S}, \mathcal{T} \subseteq 2^{\mathfrak{I}}$ as

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = [\![\mathcal{T} \diamond \mathcal{U}]\!] \tag{F.20}$$

where $\mathcal{T}, \mathcal{U}$ are some theories such that $[\![\mathcal{T}]\!] = \bigcap \mathcal{S}$ and $[\![\mathcal{U}]\!] = \bigcap \mathcal{T}$. Note that this definition is unambiguous since the existence of such $\mathcal{T}$ and $\mathcal{U}$ is guaranteed and regardless of which pair of theories with these properties we choose, we obtain the same result due to the assumption that $\diamond$ satisfies (FO4). Take some ε-based operator $\oplus$. We proceed by induction on the length n of $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$.

1° If $n = 0$, then it immediately follows that

$$[\![\Diamond \boldsymbol{T}]\!] = [\![\Diamond \langle \mathcal{T}_0 \rangle]\!] = [\![\mathcal{T}_0]\!] = [\![\bigoplus \langle \mathcal{T}_0 \rangle]\!] = [\![\bigoplus \boldsymbol{T}]\!] \enspace .$$

2° Suppose that the claim holds for n, i.e. for $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$ we have

$$[\![\Diamond \boldsymbol{T}]\!] = [\![\bigoplus \boldsymbol{T}]\!] \enspace . \tag{F.21}$$

Our goal is to show that it also holds for $n+1$, i.e. for $\boldsymbol{T}' = \langle \mathcal{T}_i \rangle_{i<n+1}$.
It follows that

$$\langle\!\langle \bigoplus \boldsymbol{T}' \rangle\!\rangle = \langle\!\langle \bigoplus \boldsymbol{T} \oplus \mathcal{T}_n \rangle\!\rangle = \{\, [\![\phi]\!] \cup \varepsilon([\![\phi]\!], \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle, \langle\!\langle \mathcal{T}_n \rangle\!\rangle) \mid \phi \in \bigoplus \boldsymbol{T} \,\} \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle \enspace .$$

Furthermore, it follows from (F.20) and (F.21) that

$$\varepsilon([\![\phi]\!], \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle, \langle\!\langle \mathcal{T}_n \rangle\!\rangle) = [\![\Diamond \boldsymbol{T} \diamond \mathcal{T}_n]\!] = [\![\Diamond \boldsymbol{T}']\!] \enspace .$$

Consequently,

$$[\![\bigoplus \boldsymbol{T}']\!] = \bigcap \langle\!\langle \bigoplus \boldsymbol{T}' \rangle\!\rangle = \bigcap \left(\{\, [\![\phi]\!] \cup [\![\Diamond \boldsymbol{T}']\!] \mid \phi \in \bigoplus \boldsymbol{T} \,\} \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle \right) \enspace .$$

In the following we show that this set is equal to $[\![\Diamond \boldsymbol{T}']\!]$. It can be equivalently written as follows:

$$\left([\![\Diamond \boldsymbol{T}']\!] \cup \bigcap \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle \right) \cap \bigcap \langle\!\langle \mathcal{T}_n \rangle\!\rangle \enspace .$$

Substituting $[\![\Diamond \boldsymbol{T}]\!]$ for $\bigcap \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle$ and distributing $\cap$ over $\cup$ yields

$$\left([\![\Diamond \boldsymbol{T}']\!] \cap [\![\mathcal{T}_n]\!] \right) \cup \left([\![\Diamond \boldsymbol{T}]\!] \cap [\![\mathcal{T}_n]\!] \right) \enspace .$$

Finally, using (FO1) and (FO2.1) we can rewrite this as

$$\begin{aligned} ([\![\Diamond \boldsymbol{T} \diamond \mathcal{T}_n]\!] \cap [\![\mathcal{T}_n]\!]) \cup [\![(\Diamond \boldsymbol{T}) \cup \mathcal{T}_n]\!] &= [\![\Diamond \boldsymbol{T} \diamond \mathcal{T}_n]\!] \cup [\![(\Diamond \boldsymbol{T}) \cup \mathcal{T}_n]\!] \\ &= [\![\Diamond \boldsymbol{T} \diamond \mathcal{T}_n]\!] = [\![\Diamond \boldsymbol{T}']\!] \enspace . \end{aligned}$$

□

F.2.2 Formula-Based Update Operators

The set of possible remainders has a number of important properties from which properties of specific formula-based operators follow. We start with two auxiliary results which make it possible to construct a subset of a theory with important properties on the semantic level.

Lemma F.51. *Let $\mathcal{T}, \mathcal{U}$ be theories. Then $\mathcal{U}$ is consistent if and only if* $\mathsf{rem}(\mathcal{T}, \mathcal{U})$ *is non-empty.*

Proof. First suppose that $\mathcal{U}$ is consistent and let $\mathcal{R}$ be the set of all subsets $\mathcal{T}'$ of $\mathcal{T}$ such that $\mathcal{T}' \cup \mathcal{U}$ is consistent. $\mathcal{R}$ must be non-empty because $\emptyset$ clearly belongs to $\mathcal{R}$. So take some subset-maximal element $\mathcal{T}^*$ of $\mathcal{R}$. It is not difficult to see that $\mathcal{T}^*$ belongs to $\mathsf{rem}(\mathcal{T}, \mathcal{U})$.

On the other hand, if $\mathsf{rem}(\mathcal{T}, \mathcal{U})$ is non-empty, then it contains some set $\mathcal{T}'$ such that $\mathcal{T}' \cup \mathcal{U}$ is consistent. Thus it follows directly that $\mathcal{U}$ is also consistent. □

Lemma F.52. *Let $\mathcal{T}, \mathcal{S}$ be theories such that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \rangle\!\rangle^{\mathfrak{I}}$, $\mathcal{T}'$ a subset of $\mathcal{T}$ and*

$$\mathcal{S}' = \left\{ \phi \in \mathcal{S} \;\middle|\; [\![\phi]\!] \in \langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} \right\} \enspace .$$

Then $\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}}$.

Proof. Suppose first that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}}$. Then it also belongs to $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}}$, so by our assumption either $\mathcal{M} = \mathfrak{I}$ or $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{S} \rangle\!\rangle$. In the former case $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}}$ and we are finished. In the latter case there is a sentence $\phi \in \mathcal{S}$ such that $[\![\phi]\!] = \mathcal{M}$ and ϕ belongs to $\mathcal{S}'$ by its definition. Consequently, $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}}$.

As for the other inclusion, if $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}}$, then either $\mathcal{M} = \mathfrak{I}$ or for some sentence $\phi \in \mathcal{S}'$ we have $\mathcal{M} = [\![\phi]\!]$. Therefore, $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}}$ by the definition of $\mathcal{S}'$. □

Lemma F.53. *Let $\mathcal{T}, \mathcal{S}, \mathcal{U}, \mathcal{V}$ be theories such that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$, $\mathcal{T}'$ a subset of $\mathcal{T}$ such that $\mathcal{T}' \cup \mathcal{U}$ is consistent and*

$$\mathcal{S}' = \left\{ \phi \in \mathcal{S} \;\middle|\; [\![\phi]\!] \in \langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} \right\} \enspace .$$

Then $\mathcal{S}' \cup \mathcal{V}$ is consistent.

Proof. To verify that $\mathcal{S}' \cup \mathcal{V}$ is consistent, we only need to use Lemma F.52 and observe that

$$\begin{aligned} [\![\mathcal{S}' \cup \mathcal{V}]\!] &= \bigcap \langle\!\langle \mathcal{S}' \cup \mathcal{V} \rangle\!\rangle = \bigcap (\langle\!\langle \mathcal{S}' \rangle\!\rangle \cup \langle\!\langle \mathcal{V} \rangle\!\rangle) = \bigcap (\langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle \mathcal{V} \rangle\!\rangle) \\ &= \bigcap (\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle) = \bigcap (\langle\!\langle \mathcal{T}' \rangle\!\rangle \cup \langle\!\langle \mathcal{U} \rangle\!\rangle) = \bigcap \langle\!\langle \mathcal{T}' \cup \mathcal{U} \rangle\!\rangle \\ &= [\![\mathcal{T}' \cup \mathcal{U}]\!] \enspace . \end{aligned}$$

□

The following result pinpoints an important property of the set of possible remainders: their syntax-independence w.r.t. individual sentences in the argument theories. More formally:

Proposition F.54 (Syntax-Independence of Remainders)**.** *Let $\mathcal{T}$, $\mathcal{S}$, $\mathcal{U}$, $\mathcal{V}$ be theories such that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$. Then,*

$$(\!(\mathsf{rem}(\mathcal{T}, \mathcal{U}))\!)^{\mathfrak{I}} = (\!(\mathsf{rem}(\mathcal{S}, \mathcal{V}))\!)^{\mathfrak{I}} \enspace .$$

Proof. We prove that $(\!(\mathsf{rem}(\mathcal{T}, \mathcal{U}))\!)^{\mathfrak{I}} \subseteq (\!(\mathsf{rem}(\mathcal{S}, \mathcal{V}))\!)^{\mathfrak{I}}$, the other inclusion follows by the same arguments since the formulation of the proposition is symmetric.

Take some $\mathcal{T}'$ from $\mathsf{rem}(\mathcal{T}, \mathcal{U})$ and put

$$\mathcal{S}' = \left\{ \phi \in \mathcal{S} \;\middle|\; [\![\phi]\!] \in \langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} \right\} \enspace .$$

We need to show that $\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}}$ belongs to $(\!(\mathsf{rem}(\mathcal{S}, \mathcal{V}))\!)^{\mathfrak{I}}$. Due to Lemma F.52, $\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}}$, so it suffices to prove that $\mathcal{S}'$ belongs to $\mathsf{rem}(\mathcal{S}, \mathcal{V})$. First, note that $\mathcal{S}'$ is clearly a subset of $\mathcal{S}$ and by Lemma F.53, $\mathcal{S}' \cup \mathcal{V}$ is consistent. We prove that $\mathcal{S}'$ is subset-maximal with these properties by contradiction. Suppose that $\mathcal{S}^*$ is such that $\mathcal{S}' \subsetneq \mathcal{S}^* \subseteq \mathcal{S}$ and $\mathcal{S}^* \cup \mathcal{V}$ is consistent and let

$$\mathcal{T}^* = \left\{ \phi \in \mathcal{T} \;\middle|\; [\![\phi]\!] \in \langle\!\langle \mathcal{S}^* \rangle\!\rangle^{\mathfrak{I}} \right\} \enspace .$$

Clearly, $\mathcal{T}^*$ is a subset of $\mathcal{T}$ and by Lemma F.53, $\mathcal{T}^* \cup \mathcal{U}$ is consistent. To reach a conflict, we need to show that $\mathcal{T}'$ is a proper subset of $\mathcal{T}^*$. First note that $\langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}}$ cannot be equal to $\langle\!\langle \mathcal{S}^* \rangle\!\rangle^{\mathfrak{I}}$ – if it were, then for every sentence $\phi \in \mathcal{S}^*$ it would hold that ϕ belongs to $\mathcal{S}$ and $[\![\phi]\!]$ belongs to $\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}}$, so ϕ belongs to $\mathcal{S}'$ by its definition, contrary to the assumption that $\mathcal{S}'$ is a proper subset of $\mathcal{S}^*$. This, together with Lemma F.52, implies that

$$\langle\!\langle \mathcal{T}' \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S}' \rangle\!\rangle^{\mathfrak{I}} \subsetneq \langle\!\langle \mathcal{S}^* \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{T}^* \rangle\!\rangle^{\mathfrak{I}} \enspace . \tag{F.22}$$

It now immediately follows that $\mathcal{T}' \neq \mathcal{T}^*$. Furthermore, for any sentence ϕ from $\mathcal{T}'$, ϕ belongs to $\mathcal{T}$ and it follows from (F.22) that $[\![\phi]\!]$ belongs to $\langle\!\langle \mathcal{S}^* \rangle\!\rangle^{\mathcal{I}}$, so ϕ belongs to $\mathcal{T}^*$ by its definition. This means that $\mathcal{T}'$ is a proper subset of $\mathcal{T}^*$, contrary to the assumption that $\mathcal{T}'$ belongs to $\mathsf{rem}(\mathcal{T},\mathcal{U})$. □

Furthermore, it follows from the maximality of possible remainders that equivalent sentences are either included in them together, or not at all.

Lemma F.55 (Equivalent Sentences in Remainders)**.** *Let $\mathcal{T}, \mathcal{U}$ be theories, $\phi, \psi \in \mathcal{T}$ sentences such that $[\![\phi]\!] = [\![\psi]\!]$ and $\mathcal{T}' \in \mathsf{rem}(\mathcal{T},\mathcal{U})$ a possible remainder. Then ϕ belongs to $\mathcal{T}'$ if and only if ψ belongs to $\mathcal{T}'$.*

Proof. Without loss of generality, assume that ϕ belongs to $\mathcal{T}'$ but ψ does not. Then $\mathcal{T}' \cup \{ \psi \}$ is a subset of $\mathcal{T}$ that is consistent with $\mathcal{U}$. This is in conflict with the maximality of $\mathcal{T}'$. □

This directly leads to the following identity.

Corollary F.56. *Let $\mathcal{T}, \mathcal{U}$ be theories, $\mathcal{R} \subseteq \mathsf{rem}(\mathcal{T},\mathcal{U})$ a set of possible remainders. Then,*

$$\bigcap (\!(\mathcal{R})\!) = \left\langle\!\!\left\langle \bigcap \mathcal{R} \right\rangle\!\!\right\rangle \;.$$

Proof. First suppose that $\mathcal{M}$ belongs to $\bigcap (\!(\mathcal{R})\!)$ and take some $\mathcal{T}' \in \mathcal{R}$ and some sentence $\phi \in \mathcal{T}'$ such that $[\![\phi]\!] = \mathcal{M}$. Now take an arbitrary $\mathcal{T}^* \in \mathcal{R}$. Since $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{T}^* \rangle\!\rangle$, there must exist a sentence $\psi \in \mathcal{T}^*$ such that $[\![\psi]\!] = \mathcal{M}$. Consequently, $[\![\phi]\!] = [\![\psi]\!]$ and by Lemma F.55 we obtain that ϕ also belongs to $\mathcal{T}^*$. Thus, ϕ belongs to $\bigcap \mathcal{R}$ and $\mathcal{M}$ belongs to $\langle\!\langle \bigcap \mathcal{R} \rangle\!\rangle$.

On the other hand, if $\mathcal{M}$ belongs to $\langle\!\langle \bigcap \mathcal{R} \rangle\!\rangle$, then there is a sentence $\phi \in \bigcap \mathcal{R}$ such that $[\![\phi]\!] = \mathcal{M}$. Consequently, $\mathcal{M}$ belongs to all members of $(\!(\mathcal{R})\!)$, thus also belongs to their intersection. □

Corollary F.57. *Let $\mathcal{T}, \mathcal{U}$ be theories, $\mathcal{R} \subseteq \mathsf{rem}(\mathcal{T},\mathcal{U})$ a set of possible remainders. Then,*

$$\bigcap (\!(\mathcal{R})\!)^{\mathcal{I}} = \left\langle\!\!\left\langle \bigcap \mathcal{R} \right\rangle\!\!\right\rangle^{\mathcal{I}} \;.$$

Proof. Follows from Corollary F.56 and from the fact that $\mathcal{I}$ belongs to both sides of the equation. □

Proposition F.58 (Properties of the WIDTIO Operator)**.** *The WIDTIO operator satisfies* (F1)*,* (F2.1) *and* (F4)*.*

Proof. By definition $\mathcal{U} \subseteq \mathcal{T} \circ_{\mathsf{WIDTIO}} \mathcal{U}$ and (F1) is obtained by applying $\langle\!\langle \cdot \rangle\!\rangle$ to both sides of this inclusion.

In order to verify that (F2.1) holds, suppose that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{T} \circ_{\mathsf{WIDTIO}} \mathcal{U} \rangle\!\rangle$. Then there is some sentence ϕ from $\mathcal{U} \cup \bigcap \mathsf{rem}(\mathcal{T},\mathcal{U})$ such that $[\![\phi]\!] = \mathcal{M}$. If ϕ belongs to $\mathcal{U}$, then it immediately follows that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle$, and consequently also to $\langle\!\langle \mathcal{T} \cup \mathcal{U} \rangle\!\rangle$. If ϕ belongs to $\mathcal{T}'$ for all $\mathcal{T}' \in \mathsf{rem}(\mathcal{T},\mathcal{U})$, then ϕ also belongs to $\mathcal{T}$. Thus, $\mathcal{M}$ is a member of $\langle\!\langle \mathcal{T} \rangle\!\rangle$, and consequently also of $\langle\!\langle \mathcal{T} \cup \mathcal{U} \rangle\!\rangle$.

Finally, to verify (F4), suppose that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$. The following follows from the definition of the WIDTIO operator, Corollary F.57 and Proposition F.54.

$$\begin{aligned}\langle\!\langle \mathcal{T} \circ_{\mathsf{WIDTIO}} \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} &= \left\langle\!\!\left\langle \mathcal{U} \cup \bigcap \mathsf{rem}(\mathcal{T},\mathcal{U}) \right\rangle\!\!\right\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} \cup \left\langle\!\!\left\langle \bigcap \mathsf{rem}(\mathcal{T},\mathcal{U}) \right\rangle\!\!\right\rangle^{\mathfrak{I}} \\ &= \langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} \cup \bigcap \langle\!\langle \mathsf{rem}(\mathcal{T},\mathcal{U}) \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}} \cup \bigcap \langle\!\langle \mathsf{rem}(\mathcal{S},\mathcal{V}) \rangle\!\rangle^{\mathfrak{I}} \\ &= \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}} \cup \left\langle\!\!\left\langle \bigcap \mathsf{rem}(\mathcal{S},\mathcal{V}) \right\rangle\!\!\right\rangle^{\mathfrak{I}} = \left\langle\!\!\left\langle \mathcal{V} \cup \bigcap \mathsf{rem}(\mathcal{S},\mathcal{V}) \right\rangle\!\!\right\rangle^{\mathfrak{I}} \\ &= \langle\!\langle \mathcal{S} \circ_{\mathsf{WIDTIO}} \mathcal{V} \rangle\!\rangle^{\mathfrak{I}} \ . \end{aligned}$$

□

Proposition F.59 (Properties of Regular Bold Operators)**.** *Regular Bold operators satisfy (F1), (F2.1) and (F4).*

Proof. By definition $\mathcal{U} \subseteq \mathcal{T} \circ^s_{\mathsf{BOLD}} \mathcal{U}$ and (F1) is obtained by applying $\langle\!\langle \cdot \rangle\!\rangle$ to both sides of this inclusion.

In order to verify that (F2.1) holds, suppose that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{T} \circ^s_{\mathsf{BOLD}} \mathcal{U} \rangle\!\rangle$. Then there is some sentence ϕ from $\mathcal{U} \cup s(\mathsf{rem}(\mathcal{T},\mathcal{U}))$ such that $[\![\phi]\!] = \mathcal{M}$. If ϕ belongs to $\mathcal{U}$, then it immediately follows that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle$, and consequently also to $\langle\!\langle \mathcal{T} \cup \mathcal{U} \rangle\!\rangle$. If ϕ belongs to $\mathcal{T}' = s(\mathsf{rem}(\mathcal{T},\mathcal{U}))$, then ϕ also belongs to $\mathcal{T}$. Thus, $\mathcal{M}$ is a member of $\langle\!\langle \mathcal{T} \rangle\!\rangle$, and consequently also of $\langle\!\langle \mathcal{T} \cup \mathcal{U} \rangle\!\rangle$.

Finally, to verify (F4), suppose that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$. The following follows from the definition of the WIDTIO operator, Proposition F.54 and the regularity property of $\circ^s_{\mathsf{BOLD}}$.

$$\begin{aligned}\langle\!\langle \mathcal{T} \circ^s_{\mathsf{BOLD}} \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} &= \langle\!\langle \mathcal{U} \cup s(\mathsf{rem}(\mathcal{T},\mathcal{U})) \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle s(\mathsf{rem}(\mathcal{T},\mathcal{U})) \rangle\!\rangle^{\mathfrak{I}} \\ &= \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle s(\mathsf{rem}(\mathcal{S},\mathcal{V})) \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \cup s(\mathsf{rem}(\mathcal{S},\mathcal{V})) \rangle\!\rangle^{\mathfrak{I}} \\ &= \langle\!\langle \mathcal{S} \circ^s_{\mathsf{BOLD}} \mathcal{V} \rangle\!\rangle^{\mathfrak{I}} \ . \end{aligned}$$

□

Proposition F.60 (Properties of the Cross-Product Operator)**.** *The Cross-Product operator satisfies (F1), (FO2.1) and (F4) but does not satisfy (F2.1).*

Proof. By definition $\mathcal{U} \subseteq \mathcal{T} \circ_{\mathsf{CP}} \mathcal{U}$ and (F1) is obtained by applying $\langle\!\langle \cdot \rangle\!\rangle$ to both sides of this inclusion.

To see that $\circ_{\mathsf{CP}}$ does not satisfy (F2.1), note that

$$\{ p, q \} \circ_{\mathsf{CP}} \{ \neg p \lor \neg q \} = \{ p \lor q, \neg p \lor \neg q \}$$

and $[\![p \lor q]\!]$ does not belong to $\langle\!\langle \{ p, q, \neg p \lor \neg q \} \rangle\!\rangle$.

In order to verify (FO2.1), take some I from $[\![\mathcal{T} \cup \mathcal{U}]\!]$. We need to show that I is a model of $\mathcal{T} \circ_{\mathsf{CP}} \mathcal{U}$. Obviously, I is a model of $\mathcal{U}$, so it remains to prove that I is a model of the sentence

$$\psi = \bigvee_{\mathcal{T}' \in \mathsf{rem}(\mathcal{T},\mathcal{U})} \ \bigwedge_{\phi \in \mathcal{T}'} \phi \ .$$

Since I is a model of $\mathcal{U}$, we conclude that $\mathcal{U}$ is consistent, so according to Lemma F.51, $\mathsf{rem}(\mathcal{T},\mathcal{U})$ is non-empty. Take some $\mathcal{T}^*$ from $\mathsf{rem}(\mathcal{T},\mathcal{U})$. We obtain the following:

$$[\![\psi]\!] = \bigcup_{\mathcal{T}' \in \mathsf{rem}(\mathcal{T},\mathcal{U})} [\![\mathcal{T}']\!] \supseteq [\![\mathcal{T}^*]\!] \supseteq [\![\mathcal{T}]\!] \ .$$

Hence since I belongs to $[\![\mathcal{T}]\!]$, it also belongs to $[\![\psi]\!]$.

Finally, to verify (F4), suppose that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{S} \rangle\!\rangle^{\mathfrak{I}}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$ and take some $\mathcal{M}$ from $\langle\!\langle \mathcal{T} \circ_{\mathsf{CP}} \mathcal{U} \rangle\!\rangle^{\mathfrak{I}}$. In the trivial case when $\mathcal{M} = \mathfrak{I}$ it immediately follows that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{S} \circ_{\mathsf{CP}} \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$. Otherwise, there is a sentence ϕ from $\mathcal{U} \cup \{ \psi \}$ such that $[\![\phi]\!] = \mathcal{M}$. If ϕ belongs to $\mathcal{U}$, then $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{U} \rangle\!\rangle$ and by assumption also to $\langle\!\langle \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$. By (F1) we then obtain that $\mathcal{M}$ belongs to $\langle\!\langle \mathcal{S} \circ_{\mathsf{CP}} \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$. On the other hand, if ϕ is ψ, then due to Proposition F.54, $(\!(\mathsf{rem}(\mathcal{T}, \mathcal{U}))\!)^{\mathfrak{I}} = (\!(\mathsf{rem}(\mathcal{S}, \mathcal{V}))\!)^{\mathfrak{I}}$, so

$$[\![\phi]\!] = [\![\psi]\!] = \bigcup_{\mathcal{T}' \in \mathsf{rem}(\mathcal{T}, \mathcal{U})} [\![\mathcal{T}']\!] = \bigcup_{\mathcal{S}' \in \mathsf{rem}(\mathcal{S}, \mathcal{V})} [\![\mathcal{S}']\!] = [\![\psi']\!]$$

where $\mathcal{S} \circ_{\mathsf{CP}} \mathcal{V} = \mathcal{V} \cup \{ \psi' \}$. Therefore, $[\![\phi]\!]$ belongs to $\langle\!\langle \mathcal{S} \circ_{\mathsf{CP}} \mathcal{V} \rangle\!\rangle^{\mathfrak{I}}$. The proof of the other inclusion is symmetric. □

Proposition 8.16. *The WIDTIO and regular Bold operators satisfy (F1), (F2.1) and (F4). The Cross-Product operator satisfies (F1), (FO2.1) and (F4) but does not satisfy (F2.1).*

Proof. Follows from Propositions F.58, F.59 and F.60. □

Proposition F.61. *If $\circ$ is an update operator that satisfies (F1), (F2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of theories $\boldsymbol{T}$, $\langle\!\langle \bigcirc \boldsymbol{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle^{\mathfrak{I}}$.*

Proof. Let the exception function ε be defined for all sets of interpretations $\mathcal{M}$ and all sets of sets of interpretations $\mathcal{S}, \mathcal{T}$ as

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = \begin{cases} \emptyset & \mathcal{M} \in \langle\!\langle \mathcal{T} \circ \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} \ ; \\ \mathfrak{I} & \mathcal{M} \notin \langle\!\langle \mathcal{T} \circ \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} \ , \end{cases}$$

where $\mathcal{T}, \mathcal{U}$ are some theories such that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \mathcal{S} \cup \{ \mathfrak{I} \}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \mathcal{T} \cup \{ \mathfrak{I} \}$. Note that this definition is unambiguous since the existence of such $\mathcal{T}$ and $\mathcal{U}$ is guaranteed and regardless of which pair of theories with these properties we choose, we obtain the same result due to the assumption that $\circ$ satisfies (F4). Take some ε-based operator $\oplus$. We proceed by induction on the length n of $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$.

1° If $n = 0$, then it immediately follows that

$$\langle\!\langle \Diamond \boldsymbol{T} \rangle\!\rangle = \langle\!\langle \Diamond \langle \mathcal{T}_0 \rangle \rangle\!\rangle = \langle\!\langle \mathcal{T}_0 \rangle\!\rangle = \langle\!\langle \bigoplus \langle \mathcal{T}_0 \rangle \rangle\!\rangle = \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle \ . \tag{F.23}$$

2° Suppose that the claim holds for n, i.e. for $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$ we have

$$\langle\!\langle \Diamond \boldsymbol{T} \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle^{\mathfrak{I}} \ . \tag{F.24}$$

Our goal is to show that it also holds for $n + 1$, i.e. for $\boldsymbol{T}' = \langle \mathcal{T}_i \rangle_{i<n+1}$. It follows that

$$\langle\!\langle \bigoplus \boldsymbol{T}' \rangle\!\rangle = \langle\!\langle \bigoplus \boldsymbol{T} \oplus \mathcal{T}_n \rangle\!\rangle = \{ [\![\phi]\!] \cup \varepsilon([\![\phi]\!], \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle, \langle\!\langle \mathcal{T}_n \rangle\!\rangle) \mid \phi \in \bigoplus \boldsymbol{T} \} \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle \ .$$

Thus $\langle\!\langle \bigoplus \boldsymbol{T}' \rangle\!\rangle^{\mathfrak{I}}$ is the same as

$$\{ [\![\phi]\!] \cup \varepsilon([\![\phi]\!], \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle, \langle\!\langle \mathcal{T}_n \rangle\!\rangle) \mid \phi \in \bigoplus \boldsymbol{T} \} \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle \cup \{ \mathfrak{I} \}$$

which in turn can be written as

$$\left\{ \mathcal{M} \,\middle|\, \mathcal{M} \in \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle \cap \langle\!\langle \bigoplus \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \right\} \\ \cup \left\{ \mathfrak{I} \,\middle|\, \mathcal{M} \in \langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle \setminus \langle\!\langle \bigoplus \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \right\} \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle \cup \{ \mathfrak{I} \}$$

and simplified into

$$\left(\langle\!\langle \bigoplus \boldsymbol{T} \rangle\!\rangle^{\mathfrak{I}} \cap \langle\!\langle \bigoplus \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \right) \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \enspace .$$

Since $\circ$ satisfies (F4), it follows from (F.24) that $\langle\!\langle \bigoplus \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} = \langle\!\langle \bigcirc \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}}$ and thus we obtain the set

$$\left(\langle\!\langle \bigcirc \boldsymbol{T} \rangle\!\rangle^{\mathfrak{I}} \cap \langle\!\langle \bigcirc \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \right) \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}}$$

and by distributing $\cup$ over $\cap$ and using (F1) and (F2.1) we obtain

$$\begin{aligned} \left(\langle\!\langle \bigcirc \boldsymbol{T} \rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \right) \cap \left(\langle\!\langle \bigcirc \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \cup \langle\!\langle \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \right) &= \langle\!\langle \bigcirc \boldsymbol{T} \cup \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \cap \langle\!\langle \bigcirc \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \\ &= \langle\!\langle \bigcirc \boldsymbol{T} \circ \mathcal{T}_n \rangle\!\rangle^{\mathfrak{I}} \\ &= \langle\!\langle \bigcirc \boldsymbol{T}' \rangle\!\rangle^{\mathfrak{I}} \enspace . \end{aligned}$$

□

Proposition F.62. *If $\circ$ an update operator that satisfies (F1), (FO2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all theories $\mathcal{T}$, $\mathcal{U}$, $[\![\mathcal{T} \circ \mathcal{U}]\!] = [\![\mathcal{T} \oplus \mathcal{U}]\!]$.*

Proof. Let the exception function ε be defined for all sets of interpretations $\mathcal{M}$ and all sets of sets of interpretations $\mathcal{S}, \mathcal{T}$ as

$$\varepsilon(\mathcal{M}, \mathcal{S}, \mathcal{T}) = [\![\mathcal{T} \circ \mathcal{U}]\!] \enspace ,$$

where $\mathcal{T}, \mathcal{U}$ are some theories such that $\langle\!\langle \mathcal{T} \rangle\!\rangle^{\mathfrak{I}} = \mathcal{S} \cup \{ \mathfrak{I} \}$ and $\langle\!\langle \mathcal{U} \rangle\!\rangle^{\mathfrak{I}} = \mathcal{T} \cup \{ \mathfrak{I} \}$. Note that this definition is unambiguous since the existence of such $\mathcal{T}$ and $\mathcal{U}$ is guaranteed and regardless of which pair of theories with these properties we choose, we obtain the same result due to the assumption that $\circ$ satisfies (F4).

Take some ε-based operator $\oplus$. Then $[\![\mathcal{T} \oplus \mathcal{U}]\!]$ is the same as

$$\bigcap (\{ [\![\phi]\!] \cup \varepsilon([\![\phi]\!], \langle\!\langle \mathcal{T} \rangle\!\rangle, \langle\!\langle \mathcal{U} \rangle\!\rangle) \mid \phi \in \mathcal{T} \} \cup \langle\!\langle \mathcal{U} \rangle\!\rangle)$$

which in turn can be written as

$$\bigcap \{ [\![\phi]\!] \cup [\![\mathcal{T} \circ \mathcal{U}]\!] \mid \phi \in \mathcal{T} \} \cap \bigcap \langle\!\langle \mathcal{U} \rangle\!\rangle$$

and simplified into

$$([\![\mathcal{T} \circ \mathcal{U}]\!] \cup [\![\mathcal{T}]\!]) \cap [\![\mathcal{U}]\!] \enspace .$$

Furthermore, due to (F1) and (FO2.1),

$$\begin{aligned} ([\![\mathcal{T} \circ \mathcal{U}]\!] \cup [\![\mathcal{T}]\!]) \cap [\![\mathcal{U}]\!] &= ([\![\mathcal{T} \circ \mathcal{U}]\!] \cap [\![\mathcal{U}]\!]) \cup ([\![\mathcal{T}]\!] \cap [\![\mathcal{U}]\!]) \\ &= [\![\mathcal{T} \circ \mathcal{U}]\!] \cup [\![\mathcal{T} \cup \mathcal{U}]\!] \\ &= [\![\mathcal{T} \circ \mathcal{U}]\!] \enspace . \end{aligned}$$

□

Theorem 8.17. *If $\circ$ is an update operator that satisfies (F1), (F2.1) and (F4), then there exists* *an exception function ε such that for every ε-based update operator $\oplus$ and all finite sequences of theories $\boldsymbol{T}$, $[\![\bigcirc \boldsymbol{T}]\!] = [\![\bigoplus \boldsymbol{T}]\!]$.*

If $\circ$ an update operator that satisfies (F1), (FO2.1) and (F4), then there exists an exception function ε such that for every ε-based update operator $\oplus$ and all theories $\mathcal{T}, \mathcal{U}$, $[\![\mathcal{T} \circ \mathcal{U}]\!] = [\![\mathcal{T} \oplus \mathcal{U}]\!]$.

Proof. Follows from Propositions F.61 and F.62. □

2012

Updates of Hybrid Knowledge Bases

Martin Slota