Artificial Intelligence 229 (2015) 33–104

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

On updates of hybrid knowledge bases composed of ontologies and rules ☆

Martin Slota *, João Leite **, Theresa Swift *

NOVA LINCS & Departamento de Informática, Universidade Nova de Lisboa, Quinta da Torre, 2829-516 Caparica, Portugal

A R T I C L E   I N F O

Article history:
Received 22 January 2014
Received in revised form 27 July 2015
Accepted 30 July 2015
Available online 4 August 2015

Keywords:
Hybrid knowledge bases
Ontologies
Description logics
Logic programs
Rules
Stable models
Answer sets
Belief change
Update

A B S T R A C T

Throughout the last decade, two distinct knowledge representation paradigms have been standardised to capture rich metadata on the Web: *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.

Instead of the usual static view of hybrid knowledge, in this paper we address its *dynamics* and in particular focus on *updates*. We develop two hybrid update semantics 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, and based on a generalised notion of *splitting*, known from Logic Programming, 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. We examine their fundamental formal properties and show that despite the different ideas behind each of them, they are fully compatible with one another, i.e. when both are applicable, they lead to the same result.

1. Introduction

Recent standardisation efforts gave rise to widely accepted knowledge representation languages such as the Web Ontology Language (OWL)[1] and Rule Interchange Format (RIF),[2] based on Description Logics [11] and Logic Programming [22,42,61,74], respectively. This has fostered a large number of ontologies and rule bases with different levels of complexity and scale. Whereas description-logic based ontologies provide the logical underpinning of intelligent access and

☆ This is a combined, revised and extended version of the material presented in [83,84,91].

* Corresponding author.
** Primary corresponding author.
E-mail addresses: martin.slota@gmail.com (M. Slota), jleite@fct.unl.pt (J. Leite), theresasturn@gmail.com (T. Swift).

1 http://www.w3.org/TR/owl-overview/.
2 http://www.w3.org/2005/rules/wiki/RIF_Working_Group.

http://dx.doi.org/10.1016/j.artint.2015.07.008

information integration, rules are widely used to represent business policies, regulations and declarative guidelines about information.

Since both ontologies and rules offer important features for knowledge representation, considerable effort has been invested in identifying a unified hybrid knowledge framework where expressivity of both formalisms could be seamlessly combined. Over the years, work on hybrid knowledge bases has matured significantly and fundamental semantic as well as computational problems have been successfully addressed (see [25,52] for an overview). MKNF Knowledge Bases [77] form one of the most mature proposals. MKNF knowledge bases provide a tight integration of the two paradigms, allowing predicates to be defined concurrently in the ontology and using rules, while at the same time being faithful to the semantics both of their ontology component and their rules component. Further, they have thoroughly examined decidability and complexity properties including a tractable variant of the formalism based on the well-founded semantics [41,59].

While such formalisms make it possible to seamlessly combine rules and ontologies in a single unified framework, they do not take into account the *dynamic character* of application areas where they are to be used. More particularly, the essential support for keeping a hybrid knowledge base *up to date*, by incorporating new and possibly conflicting information, is still missing. Nonetheless, this topic has been extensively addressed in the context of both Description Logics and Logic Programs, when taken separately.

The formal underpinning of ontology updates originates in the ample area of belief change [3], more particularly in the principles and operators used to deal with change in action theories and relational databases with incomplete information [56,57,95,96]. An *update* is generally understood as a change operation that consists in bringing a knowledge base up to date when the *world described by it changes* [56]. More technically, update operators are based on the idea that the models of a knowledge base correspond to possible states of the represented world. When a change in the world needs to be recorded, inertia is applied to each of these possible states, making only the smallest necessary modifications to reflect the change, arriving at a new collection of possible states that represent the world after the update. Later, these ideas, and particularly Winslett's update operator [57,95], were applied to perform updates of Description Logic ontologies [15,27–29,36,58,73].

When updates were tackled in the context of Logic Programming, it was only natural to consider adapting the classical update postulates and operators to deal with them. However, this led to counterintuitive results because simply applying inertia to the current stable model, or even a richer semantic characterisation, such as SE-models [93], results in the loss of essential relationships between literals that are encoded in the rules [66,85,90]. Although state-of-the-art approaches to rule updates are guided by the same basic intuitions and aspirations as ontology updates, they build upon fundamentally different principles and methods.

Many are based on the *causal rejection principle* [6,7,37,66,78], while others employ syntactic transformations and other methods, such as abduction [80], forgetting [98], prioritisation [97], preferences [31], or dependencies on defeasible assumptions [82]. Inertia, instead of being applied to the current state, is applied to *rules*. Furthermore, rather than operating on the models of a logic program, rule update semantics refer to its syntactic structure: the individual rules and, in many cases, also the literals in heads and bodies of these rules. These properties render them seemingly irreconcilable with ontology updates where *state inertia* and *syntax-independence* are central.

In order to define *generic* hybrid update operators i.e., update operators that can deal with arbitrary hybrid knowledge bases, these apparently irreconcilable approaches to dealing with knowledge updates must first be integrated, both semantically and computationally. Despite the recent developments, which already provide a narrow bridge between ontology and rule updates [87–89], this problem is still far from having a suitable solution.

In Section 3 we take an important first step to defining a generic hybrid update operator by following a different approach. In particular, 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 does not change or can be maintained manually, and can be overridden by ontology updates when necessary. For illustrative purposes, consider the following simple scenario:

Example 1 *(Electronic marketplace).* An electronic marketplace where agents sell and exchange items, resources and services needs to keep track of all the active users and sellers, available offers, product categorisation, etc. Though most of this information is kept within a Description Logic ontology, rules are used where the Closed World Assumption or reasoning with exceptions is needed. A small fragment of this hybrid knowledge looks as follows:

$$\mathsf{Seller} \sqsubseteq \mathsf{User} \tag{1}$$

$$\mathsf{ProspectiveSeller} \equiv \neg\mathsf{Seller} \sqcap \exists\mathsf{RecommendedBy}.\mathsf{Seller} \tag{2}$$

$$\neg\mathsf{User}(\mathbf{x}) \leftarrow {\sim}\mathsf{User}(\mathbf{x}). \tag{3}$$

$$\mathsf{PaysServiceFee}(\mathbf{x}) \leftarrow \mathsf{Seller}(\mathbf{x}), {\sim}\mathsf{Student}(\mathbf{x}). \tag{4}$$

Ontology axioms (1) and (2) express that every Seller in the marketplace is also its User and that every individual that is currently not a seller but has been recommended by some seller is considered a ProspectiveSeller (e.g. for marketing purposes). Rule (3) expresses that the concept User is interpreted under the Closed World Assumption, i.e. it is assumed that all of the current users are already recorded in the knowledge base, so any other individual is certainly not a user. Finally, rule (4) says that all sellers not known to be students are expected to pay a service fee.

While these axioms and rules remain pretty much the same throughout the lifetime of the knowledge base, information about particular products and users is highly dynamic and can be handled by updating the part of the ABox of the ontology.

The resulting framework allows for a seamless two-way interaction between Logic Programming rules and Description Logic axioms and is significantly more expressive than other approaches to ontology updates with a firm formal underpinning but without support for rules [15,27–29,36,58,73].

However, its applicability is limited since in many cases 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 2 *(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.[3] Assessing this risk, even at a preliminary level, involves 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.

Fig. 1 shows a knowledge base, $\mathcal{K}$ which represents a simplified fragment of this domain. $\mathcal{K}$ consists of both an ontology component $\mathcal{O}$ and a program component $\mathcal{P}$ which together define attributes of a shipment such as the country of its origination, the commodity it contains, its importer and producer. In particular, the ontology contains a geographic classification (e.g., the predicate EUCountry), along with information about producers who are located in various countries (RegisteredProducer). It also contains a classification of commodities based on their harmonised tariff information (HTSChapter, HTSHeading, and HTSCode cf. http://www.usitc.gov/tata/hts). Tariff information is also present, based on the classification of commodities (TariffCharge). 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 (e.g., SuspectedBadGuy), and about whether to inspect a shipment either to check for compliance of tariff information or for food safety issues (PartialInspection and FullInspection).

In order to address scenarios such as the one described above, we define a hybrid update semantics that is parameterised by a first-order update operator and a rule update semantics. It can deal with an interesting class of hybrid knowledge bases in which the interaction between the ontology and rules is limited, but where both the ontology and rules can be updated. One way to look at this semantics is as a *modular combination* of a first-order update operator with a rule update semantics.

The main ideas for identifying this class of hybrid knowledge bases come from the splitting theorems for Logic Programs [71]. We extract the essence of this work as *Abstract Splitting Properties* to characterise modular combinations of hybrid knowledge bases as well as of update sequences of ontologies, programs and hybrid knowledge bases. Subsequently we show that for many semantics, such as MKNF models of MKNF knowledge bases, both the models assigned by Winslett's update operator to a sequence of ontologies, and the dynamic stable models assigned to a sequence of programs by a number of rule update semantics.

Ultimately, these results enable us to define an update semantics for those hybrid knowledge bases that can be split into a sequence of *ontology and rule layers* that share information. Each ontology layer is updated using a first-order update operator, each rule layer is updated using a rule update semantics, and the partial results are combined to obtain an overall model.

The main contributions of this paper are thus as follows:

- We define a semantics for performing ontology updates in the presence of static rules, parametrised by an ontology update operator.
- We reformulate the splitting properties of [71] in an abstract manner, so they can be easily formulated for any semantic framework. We also show that prominent static hybrid semantics, ontology update operators and rule update semantics all satisfy these properties.
- 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, parameterised by an ontology update operator and a rule update semantics.
- We examine the fundamental formal properties of both proposed hybrid update semantics, showing that they are faithful to the static MKNF semantics as well as to the update semantics by which they are parameterised, and that they are fully compatible with one another.

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

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

Commodity $\equiv$ ($\exists$HTSCode.$\top$)
CherryTomato $\equiv$ ($\exists$HTSCode.{ ‘07020020’ })
GrapeTomato $\equiv$ ($\exists$HTSCode.{ ‘07020010’ })
CherryTomato $\sqsubseteq$ Tomato
CherryTomato $\sqcap$ Bulk $\equiv$ ($\exists$TariffCharge.{ \$0 })
GrapeTomato $\sqcap$ Bulk $\equiv$ ($\exists$TariffCharge.{ \$40 })
CherryTomato $\sqcap$ Prepackaged $\equiv$ ($\exists$TariffCharge.{ \$50 })
GrapeTomato $\sqcap$ Prepackaged $\equiv$ ($\exists$TariffCharge.{ \$100 })
EURegisteredProducer $\equiv$ ($\exists$RegisteredProducer.EUCountry)
LowRiskEUCommodity $\equiv$ ($\exists$ExpeditableImporter.$\top$) $\sqcap$ ($\exists$CommodCountry.EUCountry)
EdibleVegetable $\equiv$ ($\exists$HTSChapter.{ ‘07’ })
Tomato $\equiv$ ($\exists$HTSHeading.{ ‘0702’ })
Tomato $\sqsubseteq$ EdibleVegetable
GrapeTomato $\sqsubseteq$ Tomato
CherryTomato $\sqcap$ GrapeTomato $\sqsubseteq \bot$
Bulk $\sqcap$ Prepackaged $\sqsubseteq \bot$

ShpmtCommod(s_1, c_1)
ShpmtImporter(s_1, i_1)
ShpmtCommod(s_2, c_2)
ShpmtImporter(s_2, i_2)
ShpmtCountry(s_2, *portugal*)
ShpmtCommod(s_3, c_3)
ShpmtImporter(s_3, i_3)
ShpmtCountry(s_3, *portugal*)
RegisteredProducer(p_1, *portugal*)
RegisteredProducer(p_2, *slovakia*)
ShpmtDeclHTSCode(s_1, ‘07020020’)
CherryTomato(c_1) Bulk(c_1)
ShpmtDeclHTSCode(s_2, ‘07020020’)
CherryTomato(c_2) Prepackaged(c_2)
ShpmtDeclHTSCode(s_3, ‘07020010’)
GrapeTomato(c_3) Bulk(c_3)
ShpmtProducer(s_3, p_1)
EUCountry(*portugal*)
EUCountry(*slovakia*)

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

AdmissibleImporter($\mathbf{x}$) $\leftarrow$ $\sim$SuspectedBadGuy($\mathbf{x}$).
SuspectedBadGuy(i_1).
ApprovedImporterOf($i_2, \mathbf{x}$) $\leftarrow$ EdibleVegetable($\mathbf{x}$).
ApprovedImporterOf($i_3, \mathbf{x}$) $\leftarrow$ GrapeTomato($\mathbf{x}$).
CommodCountry($\mathbf{x}, \mathbf{y}$) $\leftarrow$ ShpmtCommod($\mathbf{z}, \mathbf{x}$), ShpmtCountry($\mathbf{z}, \mathbf{y}$).
ExpeditableImporter($\mathbf{x}, \mathbf{y}$) $\leftarrow$ ShpmtCommod($\mathbf{z}, \mathbf{x}$), ShpmtImporter($\mathbf{z}, \mathbf{y}$), AdmissibleImporter($\mathbf{y}$), ApprovedImporterOf($\mathbf{y}, \mathbf{x}$).
CompliantShpmt($\mathbf{x}$) $\leftarrow$ ShpmtCommod($\mathbf{x}, \mathbf{y}$), HTSCode($\mathbf{y}, \mathbf{z}$), ShpmtDeclHTSCode($\mathbf{x}, \mathbf{z}$).
RandomInspection($\mathbf{x}$) $\leftarrow$ ShpmtCommod($\mathbf{x}, \mathbf{y}$), Random($\mathbf{y}$).
PartialInspection($\mathbf{x}$) $\leftarrow$ RandomInspection($\mathbf{x}$).
PartialInspection($\mathbf{x}$) $\leftarrow$ ShpmtCommod($\mathbf{x}, \mathbf{y}$), $\sim$LowRiskEUCommodity($\mathbf{y}$).
FullInspection($\mathbf{x}$) $\leftarrow$ $\sim$CompliantShpmt($\mathbf{x}$).
FullInspection($\mathbf{x}$) $\leftarrow$ ShpmtCommod($\mathbf{x}, \mathbf{y}$), Tomato($\mathbf{y}$), ShpmtCountry($\mathbf{x}$, *slovakia*).

Fig. 1. MKNF knowledge base for Cargo Imports.

- We show how the semantics can be applied to properly deal with non-trivial updates in the scenarios from Examples 1 and 2.

The remainder of this paper is structured as follows: We present the necessary theoretical background in Section 2 while in Section 3 we define the semantics for ontology updates in the presence of static rules, examine its properties and illustrate its use in the context of Example 1. In Section 4 we formulate the Abstract Splitting Properties and show that a variety of semantics satisfy them. Based on these results, in Section 5 we define another hybrid update semantics that modularly combines a given first-order update operator and a rule update semantics, analyse its formal properties and show how it can deal with non-trivial updates of the hybrid knowledge base in Example 2. Finally, we discuss our results and conclude in Section 7. The proofs of all formal results are presented in Appendices A–D.

2. Background

2.1. First-order logic and description logics

We first briefly summarise aspects of first-order logic that are relevant to the theoretical results that follow, along with special assumptions that are made throughout the paper. We assume a first-order language, $\mathcal{L}$ that will form the basis for representing both ontological and rule-based knowledge. $\mathcal{L}$ is function-free apart from constant functions so that first-order *atomic predicates*, *formulas*, and *sentences*, defined inductively over disjoint sets of *constant symbols* $\mathcal{C}$, *predicate symbols* $\mathcal{P}$, along with a countably infinite set of variable symbols. A first-order formula is *ground* if it contains no variables. The set of all first-order sentences is denoted by Φ. A *first-order theory* is a set of first-order sentences.

An interpretation is a set of atoms. Throughout the paper we adopt the *Standard Name Assumption* about interpretations from [77].

Table 1
Interpretation of $\mathcal{ALCIO}$ expressions (under the Standard Names Assumption).

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	$\{ a \}$	$\{ a \}$

Definition 3. A first-order interpretation I over $\mathcal{L}$ employs the Standard Name Assumption if

- Condition 1: The universe $\mathcal{L}_U$ of I contains all constant symbols in $\mathfrak{C}$ along with a countably infinite number of additional constants called *parameters*, $\mathfrak{C}_{Param}$.
- Condition 2: $c^I = c$ for each $c \in \mathfrak{C}$.
- Condition 3: The predicate $\approx$ is interpreted in I as a congruence relation – that is, it is reflexive, symmetric, transitive, and it allows the replacement of equals by equals.

If I employs the Standard Name Assumption, we term it an *SNA interpretation*, and we denote the set of all SNA interpretations of $\mathcal{L}$ as $\mathfrak{I}_{\mathcal{L}}$.

Using condition 3 of Definition 3 allows us to introduce the relation $\equiv$ in ontologies by treating it as a congruence relation rather than as true equality (i.e., syntactic identity). However condition 2 also implies that any interpretation that satisfies the Standard Names Assumption is a Herbrand interpretation for $\mathcal{L}$. As will be discussed below, this assumption will support the logic programming semantics that will be used for rules. At the same time, Proposition 3.2 of [77] shows that any first-order formula is satisfiable if and only if it is satisfiable in a model that employs the Standard Name Assumption.

We denote the set of all SNA interpretations that are models of a first-order theory $\mathcal{T}$ by $[\![\mathcal{T}]\!]$ and we write $I \models \mathcal{T}$ whenever I belongs to $[\![\mathcal{T}]\!]$. 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 $[\![\mathcal{T}]\!] = [\![\mathcal{S}]\!]$. The models, entailment and equivalence are specialised to first-order sentences by treating every sentence ϕ as the theory $\{ \phi \}$.

2.1.1. Description logics

Description Logics (DLs) are specialised logics for which the standard reasoning tasks, such as satisfiability and entailment, are usually decidable; in this paper attention is restricted to DLs that are in fact fragments of first-order logic. We assume a basic familiarity with DLs (cf. e.g., [11]), although we review certain aspects of DLs here. DL formulas usually are written in a variable-free syntax, which we use in this paper (e.g., in the ontology portion, $\mathcal{O}$ of Fig. 1). A DL is used to describe an *ontology*, i.e. to specify a shared conceptualisation of a domain of interest: thus (for our purposes) an ontology corresponds to a theory formulated in a fragment of first-order logic.

As a specific instance, we briefly review the syntax and direct semantics of the DL $\mathcal{ALCIO}$, an extension of the DL $\mathcal{ALC}$ (*Attributive concept Language with Complements*) with inverse roles ($\mathcal{I}$) and nominals ($\mathcal{O}$) that is used in examples throughout this work.

Definition 4 *($\mathcal{ALCIO}$ syntax).* We start with a set of atomic concept and role names.

The set of $\mathcal{ALCIO}$ *role expressions* is the smallest set containing all atomic role names, and the expression R^- for all role expressions R.

The set of $\mathcal{ALCIO}$ *concept expressions* is the smallest set containing $\top$, $\bot$, all atomic 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 $a \in \mathfrak{C}$.

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.

An $\mathcal{ALCIO}$ *ABox* is a finite set of axioms (sometimes called assertions) of the forms $C(a)$, $R(a,b)$, $\neg R(a,b)$, $a \approx b$ where C is a concept expression, R is a role name and a, b are individual names.

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

A direct semantics for $\mathcal{ALCIO}$ can be defined as in Table 1, which describes an interpretation that satisfies the Standard Names Assumption Definition 3, for role and concept expressions. Note that by part 2 of Definition 3, the interpretation of

Table 2
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	$(a^I, b^I) \notin R^I$
Ontologies		
$I \models \mathcal{O}$	if and only if	$I \models \phi$ for all $\phi \in \mathcal{O}$

a nominal expression $\{ a \}$ is simply the singleton set containing a. Table 2 then shows the direct satisfiability of $\mathcal{ALCIO}$ axioms and ontologies. The relation (role name) $\approx$ is simply a congruence relation as in Definition 3, with $a \not\approx b$ shorthand for $\neg(a \approx b)$.

Alternatively, a DL axiom of $\mathcal{ALCIO}$ can be translated to a first-order sentence, which is captured at an abstract level by the following definition.

Definition 5 *(DL semantics by translation).* Let ϕ be a DL axiom. By $\kappa(\phi)$ we denote a first-order sentence that *semantically characterises* ϕ.

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 $\kappa(\mathcal{O}) \models \kappa(\mathcal{O}')$, and that $\mathcal{O}$ *is equivalent to* $\mathcal{O}'$, denoted by $\mathcal{O} \equiv \mathcal{O}'$, if $\kappa(\mathcal{O}) \equiv \kappa(\mathcal{O}')$. 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 $\kappa(\mathcal{O}) \models \phi$.

We note that while some DLs, such as $\mathcal{ALC}$, are simply fragments of first-order logic, which are usually written in a variable-free syntax, others, including $\mathcal{ALCIO}$, can also be translated to first-order logic in a straightforward manner. Thus for most DLs, the κ translation function is quite simple, and the difference between $\kappa(\mathcal{O})$ and $\mathcal{O}$ is largely a technical distinction. This is illustrated in the following example.

Example 6. Consider the translation of DL axioms from Example 2 into sentences in first-order logic. The axiom

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

can be translated to

$$\forall x_1.(\mathit{LowRiskEUCommodity}(x_1) \approx (\exists x_2.\mathit{ExpeditableImporter}(x_1, x_2) \\ \wedge \exists x_2.(\mathit{CommodCountry}(x_1, x_2) \wedge \mathit{EUCountry}(x_2))))$$

while

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

can be translated to

$$\forall x_1.((\mathit{GrapeTomato}(x_1) \wedge \mathit{Prepackaged}(x_1)) \approx \exists x_2.\mathit{TariffCharge}(x_1, \text{`\$100'})).$$

Note that variables are not required in the DL syntax because a DL class expression corresponds to a logical formula with a very strict scoping of quantifiers. In fact, a $\mathcal{ALCIO}$ DL axiom corresponds to a logical sentence that can be expressed in nor more than 2 variables, and theories that contain only sentences are known to be decidable [81].

Unless stated otherwise, we do not constrain ourselves to $\mathcal{ALCIO}$ or to any other particular DL for representing ontologies. Our main assumption is that the semantics of the DL is given by a translation into first-order logic.

2.2. Logic programming

Similarly to 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 generalised logic programs and define the *stable model semantics* [42,43] for such programs.

Syntactically, logic programs are built from *atoms* formed over the language $\mathcal{L}$ (Section 2.1). An *objective literal* is an atom p or its (strong) negation $\neg p$. We denote the set of all objective literals by *Lits* and the set of ground objective literals by Lits_G. A *default literal* is an objective literal preceded by $\sim$ denoting *default negation*. A *literal* is either an objective literal or a default literal. Given a set of literals S, we introduce the following notation: $S^+ = \{ l \in \mathit{Lits} \mid l \in S \}$, $S^- = \{ l \in \mathit{Lits} \mid {\sim}l \in S \}$, ${\sim}S = \{ {\sim}L \mid L \in S \}$. As a convention, double default negation is absorbed, so ${\sim}{\sim}l$ denotes the objective literal l.

A *rule* is a pair $\pi = (\mathsf{H}_\pi, \mathsf{B}_\pi)$ where H_π is a literal,[4] referred to as *head of* π, and B_π is a finite set of literals, referred to as *body of* π. Given a rule $\pi = (\mathsf{H}_\pi, \mathsf{B}_\pi)$ with $\mathsf{H}_\pi = l$ and $\mathsf{B}_\pi = \{p_1, \ldots, p_m, {\sim}p_{m+1}, \ldots, {\sim}p_n\}$, for convenience, we often write π as $(\mathsf{H}_\pi \leftarrow \mathsf{B}^+_\pi, {\sim}\mathsf{B}^-_\pi.)$ or as $l \leftarrow p_1, \ldots, p_m, {\sim}p_{m+1}, \ldots, {\sim}p_n$. A rule is called *ground* if it does not contain variables; *positive* if it does not contain any default literal; *a fact* if 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. The *grounding* of a program $\mathcal{P}$ is defined as $\mathsf{ground}(\mathcal{P}) = \bigcup_{\pi \in \mathcal{P}} \mathsf{ground}(\pi)$.

An *LP interpretation* is a subset of Lits_G that does not contain both p and $\neg p$ for any ground atom p. Note that LP interpretations are (possibly partial) Herbrand interpretation so that any SNA interpretation (Definition 3) is an LP-interpretation. The following conditions define when a LP interpretation J satisfies various constructs of $\mathcal{L}$. For a given ground objective literal l, ground default literal ${\sim}l$, set of ground literals S, ground rule π, and ground program $\mathcal{P}$:

$$\begin{array}{llllll} J \models l & \text{iff} & l \in J \ , & J \models {\sim}l & \text{iff} & l \notin J \ , \\ J \models S & \text{iff} & J \models L \text{ for all } L \in S \ , & J \models \pi & \text{iff} & J \models \mathsf{H}_\pi \text{ or } J \not\models \mathsf{B}_\pi \ , \\ J \models \mathcal{P} & \text{iff} & J \models \pi \text{ for all } \pi \in \mathcal{P} \ . & & & \end{array}$$

We also say that J is a *model of* $\mathcal{P}$ if $J \models \mathcal{P}$ and that $\mathcal{P}$ is *consistent* if it has a model. Also, J is a *stable model of* $\mathcal{P}$ if J is a subset-minimal model of the *reduct of* $\mathcal{P}$ *relative to* J:

$$\mathcal{P}^J = \left\{ \mathsf{H}_\pi \leftarrow \mathsf{B}^+_\pi. \mid \pi \in \mathcal{P} \wedge J \models {\sim}\mathsf{B}^-_\pi \right\} \ .$$

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 $[\![\mathcal{P}]\!]_{\mathsf{SM}}$.[5]

2.3. Logic of minimal knowledge and negation as failure

The logic of Minimal Knowledge and Negation as Failure (MKNF) [69] was introduced with the goal of unifying several existing approaches to nonmonotonic reasoning, such as default logic, autoepistemic logic, and logic programming. It was latter adopted as the basis for MKNF Knowledge Bases [77], used to join DL knowledge bases and logic programs in a seamless way. In this section we will briefly review MKNF, and leave MKNF Knowledge Bases to be reviewed in the next section.

MKNF is an extension of first-order logic with two modal operators: **K** and **not**. *MKNF sentences* and *theories* are defined by extending function-free first-order syntax by these modal operators in a natural way. In this paper, we assume that the language used for an MKNF theory is based on the language $\mathcal{L}$ of Section 2.1.

Example 7. The theory $\mathcal{T}_{prop}$, consisting of the set with the following sentences, is an example of an MKNF theory:

$$\begin{aligned} p_1 &\supset p_2 \\ \mathbf{K}\, p_4 &\supset \mathbf{K}\, p_3 \\ \mathbf{not}\, p_6 &\supset \mathbf{K}\, p_5 \\ \mathbf{not}\, p_5 &\supset \mathbf{K}\, p_6 \end{aligned}$$

The semantics of MKNF theories is determined by special Kripke structures as follows.

Definition 8 *(MKNF satisfiability).* Let $\mathcal{T}$ be an MKNF theory, ϕ be an MKNF formula, and p an atom. An *MKNF structure* is a triple $(I, \mathcal{M}, \mathcal{N})$ where $I \in \mathfrak{I}_\mathcal{L}$ and $\mathcal{M}, \mathcal{N} \subseteq \mathfrak{I}_\mathcal{L}$.[6] Let $\phi[a/\mathbf{x}]$ denote the formula obtained from ϕ by replacing every unbound occurrence of variable $\mathbf{x}$ with the constant symbol a. Satisfiability of $\mathcal{T}$ in $(I, \mathcal{M}, \mathcal{N})$ is defined as follows.

[4] Sometimes, for convenience, we take H_π to be a singleton containing a literal.

[5] Note that, unlike in [43], we do not assign any stable models to inconsistent programs.

[6] Recall that $\mathfrak{I}_\mathcal{L}$ is the set of all SNA interpretations over $\mathcal{L}$. The original formulation in [69] was not restricted to SNA interpretations.

$$
\begin{array}{lll}
(I, \mathcal{M}, \mathcal{N}) \models p & \text{if and only if} & I \models p \\
(I, \mathcal{M}, \mathcal{N}) \models \neg\phi & \text{if and only if} & (I, \mathcal{M}, \mathcal{N}) \not\models \phi \\
(I, \mathcal{M}, \mathcal{N}) \models \phi_1 \wedge \phi_2 & \text{if and only if} & (I, \mathcal{M}, \mathcal{N}) \models \phi_1 \text{ and } (I, \mathcal{M}, \mathcal{N}) \models \phi_2 \\
(I, \mathcal{M}, \mathcal{N}) \models \exists \mathbf{x} : \phi & \text{if and only if} & (I, \mathcal{M}, \mathcal{N}) \models \phi[a/\mathbf{x}] \text{ for some } a \in (\mathcal{C} \cup \mathcal{C}_{Param}) \\
(I, \mathcal{M}, \mathcal{N}) \models \mathbf{K}\,\phi & \text{if and only if} & (J, \mathcal{M}, \mathcal{N}) \models \phi \text{ for all } J \in \mathcal{M} \\
(I, \mathcal{M}, \mathcal{N}) \models \mathbf{not}\,\phi & \text{if and only if} & (J, \mathcal{M}, \mathcal{N}) \not\models \phi \text{ for some } J \in \mathcal{N} \\
(I, \mathcal{M}, \mathcal{N}) \models \mathcal{T} & \text{if and only if} & (I, \mathcal{M}, \mathcal{N}) \models \phi \text{ for all } \phi \in \mathcal{T}
\end{array}
$$

The symbols $\top$, $\bot$, $\vee$, $\forall$ and $\supset$ are interpreted in the usual manner. Intuitively, I is used to interpret the first-order parts of an MKNF theory while $\mathcal{M}$ and $\mathcal{N}$ can be thought of as sets of possible worlds over which the **K** and **not** modalities are respectively evaluated. Note that as the size of $\mathcal{M}$ or $\mathcal{N}$ increases, the number of true formulas decreases for **K** and increases for **not**.

Definition 9 *(MKNF model).* For any $\mathcal{M} \subseteq \mathcal{I}_{\mathcal{L}}$ and any MKNF theory $\mathcal{T}$, we write $\mathcal{M} \models \mathcal{T}$ if $(I, \mathcal{M}, \mathcal{M}) \models \mathcal{T}$ for all $I \in \mathcal{M}$. An *MKNF interpretation* is a non-empty subset of $\mathcal{I}_{\mathcal{L}}$. We denote the set of all MKNF interpretations together with the empty set by $\mathbb{M}$, i.e. $\mathbb{M} = 2^{\mathcal{I}_{\mathcal{L}}}$. Given an MKNF theory $\mathcal{T}$, an MKNF interpretation $\mathcal{M}$ is an *MKNF model of* $\mathcal{T}$ if $\mathcal{M} \models \mathcal{T}$, and, for every MKNF interpretation $\mathcal{M}' \supsetneq \mathcal{M}$, there is some $I' \in \mathcal{M}'$ such that $(I', \mathcal{M}', \mathcal{M}) \not\models \mathcal{T}$.

Example 10. In order to provide a sense of how MKNF satisfiability and entailment function, we examine the validity of various formulas in this logic. In the following, ϕ is a first-order formula and $\mathcal{M} \subseteq \mathcal{I}_{\mathcal{L}}$ such that $\mathcal{M}$ is non-empty.

- $\mathcal{M} \models \phi \equiv \mathbf{K}\,\phi$, as is evident directly from Definitions 8 and 9.[7]
- $\mathcal{M} \models \mathbf{not}\,\phi \supset \mathbf{K}\,\mathbf{not}\,\phi$. Suppose $\mathcal{M} \models \mathbf{not}\,\phi$. This means that for every $I \in \mathcal{M}$, $(I, \mathcal{M}, \mathcal{M}) \models \mathbf{not}\,\phi$, which by Definition 8 implies that $\mathcal{M} \models \mathbf{K}\,\mathbf{not}\,\phi$.
- $\mathcal{M} \models \neg\phi \supset \mathbf{not}\,\phi$. Let $\mathcal{M} \models \neg\phi$. Then by Definition 8, for some $I \in \mathcal{M}$, $(I, \mathcal{M}, \mathcal{M}) \not\models \phi$, which means that $\mathcal{M} \models \mathbf{not}\,\phi$. Note that the converse does not hold. If $\mathcal{M} \models \mathbf{not}\,\phi$, then there is some $I \in \mathcal{M}$ such that $(I, \mathcal{M}, \mathcal{M}) \not\models \phi$: however for $J \in \mathcal{M}$, such that $J \neq I$, it is not necessarily the case that $(J, \mathcal{M}, \mathcal{M}) \models \neg\phi$, which must be the case, as $\neg\phi \equiv \mathbf{K}\,\neg\phi$.
- $\mathcal{M} \models \phi \supset \mathbf{not}\,\mathbf{not}\,\phi$. Since $\phi \supset \mathbf{K}\,\phi$, then for any $I \in \mathcal{M}$, ϕ holds. Accordingly, for some $J \in \mathcal{M}$, $(J, \mathcal{M}, \mathcal{M}) \not\models \mathbf{not}\,\phi$, so $\mathcal{M} \models \mathbf{not}\,\mathbf{not}\,\phi$. Note that the converse does not hold. Let $\mathcal{M}$ be such that for $I_1 \in \mathcal{M}$, $(I_1, \mathcal{M}, \mathcal{M}) \models \phi$, but for all other $I \in \phi$, $(I, \mathcal{M}, \mathcal{M}) \not\models \phi$. Then $\mathcal{M} \models \mathbf{not}\,\mathbf{not}\,\phi$, but $\mathcal{M} \not\models \phi$.

Example 11. Considering the MKNF theory of Example 7, $\mathcal{T}_{prop}$, its MKNF models $\mathcal{M}$ are subsets of $[\![p_1 \supset p_2]\!] = [\![\neg p_1]\!] \cup [\![p_2]\!]$. Additionally, they must also be subsets of either $[\![p_5]\!]$ or $[\![p_6]\!]$. Due to the maximality of MKNF models that is implicit in their definition, $\mathcal{T}_{prop}$ has exactly two MKNF models:

$$\mathcal{M}_1 = ([\![\neg p_1]\!] \cup [\![p_2]\!]) \cap [\![p_5]\!] \qquad \text{and} \qquad \mathcal{M}_2 = ([\![\neg p_1]\!] \cup [\![p_2]\!]) \cap [\![p_6]\!] \ .$$

As illustrated in the preceding example, the use of the **K** and **not** modalities in MKNF allows classical models to be combined with non-monotonic features typically present in logic programming. The example itself suggests a relationship between MKNF sentences of the form

$$\mathbf{K}\,p_1 \wedge \cdots \wedge \mathbf{K}\,p_m \wedge \mathbf{not}\,p_{m+1} \wedge \cdots \wedge \mathbf{not}\,p_n \supset \mathbf{K}\,p$$

and logic programming rules of the form:

$$p \leftarrow p_1, \ldots, p_m, \sim p_{m+1}, \ldots, \sim p_n$$

where the **K** operator captures the minimality of positive inference in logic programming, while the **not** operator captures default negation.

Additionally, it turns out that MKNF supports a number of general extensions of logic programming rules beyond the form described in Section 2.2 – for instance those whose heads contain existential variables. The relationships between MKNF, Description Logics and Logic Programming will be made clear when we introduce MKNF Knowledge Bases in the next subsection.

[7] The validity of this equivalence is a feature distinguishing MKNF from the similar logic of Minimal *Belief* with Negation as Failure [70].

2.4. MKNF knowledge bases

MKNF knowledge bases [77] consist of two components – an ontology $\mathcal{O}$ and an MKNF program $\mathcal{P}$, both formed over the language $\mathcal{L}$ of Section 2.1. Their semantics is given by translation to an MKNF theory as shown below.

The programs used in MKNF knowledge bases generalise the logic programs of Section 2.2 in the following manner. A *generalised atom* is a first-order formula. A generalised atom is *ground* if it is a sentence – if the variables it contains are explicitly bound by a quantifier. 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 $\mathcal{C}$. Note that, if p is an atomic formula, it is also a generalised atom. However, p has no quantified variables, so for atomic formulas, this new definition of grounding specialises to the definition used in Section 2.2. Other logic programming related notions are naturally extended for generalised atoms. A *generalised default literal* is a generalised atom preceded by the symbol for default negation, $\sim$. A *generalised literal* is either a generalised atom or a generalised default literal. A **K**-atom is a generalised atom preceded by the **K** modal operator. Given a set of generalised literals S, we introduce the following notation:

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

An *MKNF rule* is a pair $\pi = (\mathsf{H}_\pi, \mathsf{B}_\pi)$ where H_π is a generalised literal, referred to as *head of* π, and B_π is a finite set of generalised literals, referred to as *body of* π. As before, given an MKNF rule $\pi = (\mathsf{H}_\pi, \mathsf{B}_\pi)$ with $\mathsf{H}_\pi = \{\xi\}$ and $\mathsf{B}_\pi = \{\xi_1, \dots, \xi_m, {\sim}\xi_{m+1}, \dots, {\sim}\xi_n\}$, for convenience, we often write π as $(\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi^+, {\sim}\mathsf{B}_\pi^-.)$ or as $\xi \leftarrow \xi_1, \dots, \xi_m, {\sim}\xi_{m+1}, \dots, {\sim}\xi_n$. An MKNF rule is called *ground* if it contains only ground generalised literals; *positive* if it does not contain any generalised default literal; *a fact* if its body is empty. An *MKNF program* is a set of MKNF rules. An *MKNF knowledge base* is a pair $(\mathcal{O}, \mathcal{P})$ where $\mathcal{O}$ is an ontology and $\mathcal{P}$ is an MKNF program.[8] The set of **K**-atoms occurring in the head of some rule in $\mathcal{P}$ is denoted as $\mathrm{HA}(\mathcal{P})$.

The *grounding* of an MKNF rule π is the set of MKNF rules $\mathsf{ground}(\pi)$ obtained by replacing all free (non-quantified) variables in π with constant symbols from $\mathcal{C}$ in all possible ways. An MKNF program is *ground* if all its rules are ground; *positive* if all its rules are positive. The *grounding* of an MKNF program $\mathcal{P}$ is defined as $\mathsf{ground}(\mathcal{P}) = \bigcup_{\pi \in \mathcal{P}} \mathsf{ground}(\pi)$. An MKNF knowledge base is *ground* if $\mathcal{P}$ is ground; *positive* if $\mathcal{P}$ is positive. 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 to an MKNF theory. To this end we extend the function κ (introduced in Definition 5 to translate an ontology to a first-order theory) to translate any MKNF knowledge base $\mathcal{K}$ to an MKNF theory $\kappa(\mathcal{K})$.

Definition 12. Let ξ be a generalised atom, S as set of generalised literals, π a rule with a vector $\vec{\mathbf{x}}$ of free variables, $\mathcal{P}$ an MKNF program, $\mathcal{O}$ and ontology, and $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ an MKNF knowledge base.

$$\kappa(\xi) = \mathbf{K}\,\xi \qquad \kappa({\sim}\xi) = \mathbf{not}\,\xi$$
$$\kappa(S) = \bigwedge\{\, \kappa(L) \mid L \in S \,\} \qquad \kappa(\pi) = (\forall \vec{\mathbf{x}} : \kappa(\mathsf{B}_\pi) \supset \kappa(\mathsf{H}_\pi))$$
$$\kappa(\mathcal{P}) = \{\, \kappa(\pi) \mid \pi \in \mathcal{P} \,\} \qquad \kappa(\mathcal{K}) = \kappa(\mathcal{O}) \cup \kappa(\mathcal{P})$$

For any such syntactic unit Δ we write $\mathcal{M} \models \Delta$ whenever $\mathcal{M} \models \kappa(\Delta)$. Given an MKNF knowledge base $\mathcal{K}$, we say that an MKNF interpretation $\mathcal{M}$ is an *MKNF model of* $\mathcal{K}$ if it is an MKNF model of $\kappa(\mathcal{K})$.

Example 13. Consider the MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ where

$$\mathcal{O}: \quad p_1 \sqsubseteq p_2 \qquad\qquad \mathcal{P}: \quad \begin{aligned} &p_3 \leftarrow p_4. \\ &p_5 \leftarrow {\sim}p_6. \\ &p_6 \leftarrow {\sim}p_5. \end{aligned}$$

Then, $\kappa(\mathcal{K})$ is the MKNF theory (discussed in Examples 7 and 11):

$$\begin{aligned} p_1 &\supset p_2 \\ \mathbf{K}\,p_4 &\supset \mathbf{K}\,p_3 \\ \mathbf{not}\,p_6 &\supset \mathbf{K}\,p_5 \\ \mathbf{not}\,p_5 &\supset \mathbf{K}\,p_6 \end{aligned}$$

whose MKNF models are

$$\mathcal{M}_1 = ([\![\neg p_1]\!] \cup [\![p_2]\!]) \cap [\![p_5]\!] \qquad \text{and} \qquad \mathcal{M}_2 = ([\![\neg p_1]\!] \cup [\![p_2]\!]) \cap [\![p_6]\!] \ .$$

[8] The definition introduced here differs from that of [77] in that we do not allow disjunctions in the heads of rules.

Definitions 9 and 12 imply that an MKNF model $\mathcal{M}$ of an MKNF knowledge base is an SNA interpretation over $\mathcal{L}$. However, $\mathcal{M}$ also can be represented as a set of generalised atoms. To explain this, let $\mathcal{M}$ be an MKNF interpretation, S a set of ground **K** atoms, and define the subset of S *induced* by $\mathcal{M}$ as the set $\{\mathbf{K}\xi \in S \mid \mathcal{M} \models \xi\}$. Additionally, let J be a subset of $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$, and define *the objective knowledge of* J *with respect to* $\mathsf{ground}(\mathcal{K})$, $\mathsf{OB}_{\mathcal{O},J}$, as being a first-order theory defined as:

$$\mathsf{OB}_{\mathcal{O},J} = \{\kappa(\mathcal{O})\} \cup J.$$

Theorem 14. *(See [77].) Let $\mathcal{M}$ be a model of a ground MKNF knowledge base, $\mathsf{ground}(\mathcal{K})$, and let J be the subset of $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ that is induced by $\mathcal{M}$. Then, $\mathcal{M}$ is equal to set of first order interpretations $\mathcal{M}' = \{I \mid I \models \mathsf{OB}_{\mathcal{O},J}\}$.*

Thus, for the purposes of modelling MKNF knowledge bases, a subset of MKNF structures (Definition 8) are taken to be MKNF models when they fulfil the conditions of Definition 9. Such an MKNF model can be represented as a subset of $\mathfrak{I}_{\mathcal{L}}$. However, due to Theorem 14, it can be represented even more simply as a subset of $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ which, if $\mathcal{P}$ is a logic program with no generalised atoms, is simply an LP-interpretation. Although the results in this paper are based directly on MKNF models, the fact that such models can be represented by sets of head atoms is a critical feature to the practical usefulness of MKNF knowledge bases.

We are now ready to review some of the desirable properties of MKNF knowledge bases.

Proposition 15 *(Basic properties of MKNF knowledge bases). (See [77].)*

(Faithfulness to DLs) *Let $\mathcal{O}$ be an ontology and ϕ a first-order sentence. Then $\mathcal{O} \models \phi$ if and only if for every MKNF model $\mathcal{M}$ of the MKNF knowledge base $(\mathcal{O}, \emptyset)$ it holds that $\mathcal{M} \models \phi$.*[9]

(Faithfulness to Stable Models) *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 LP interpretation corresponding to $\mathcal{M}$ is a stable model of $\mathcal{P}$.*

(Grounding) *The MKNF models of an MKNF knowledge base $\mathcal{K}$ coincide with the MKNF models of its grounding $\mathsf{ground}(\mathcal{K})$.*

(Elimination of Default Negation in the Head) *Let $\mathcal{K}$ be an MKNF knowledge base and $\mathcal{K}'$ the MKNF knowledge base obtained from $\mathcal{K}$ by replacing every rule with default negation in the head*

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

where $\bot$ is a generalised atom representing a contradiction. The MKNF models of $\mathcal{K}$ coincide with the MKNF models of $\mathcal{K}'$.

2.4.1. Decidability of queries to MKNF knowledge bases

Given that an MKNF knowledge base, $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, combines first-order logic with recursive rules, entailment of a ground atom from $\mathcal{K}$ is not necessarily decidable even if entailment is decidable from $\mathcal{O}$ and $\mathcal{P}$ taken separately. In addition, as the model of $\mathcal{K}$ is based on sets of interpretations, the question arises of how efficiently such a model can be computed, even in cases when $\mathcal{K}$ is decidable.

These questions were addressed in [77], and we informally summarise the results here, as motivation for the work in the following sections. At an informal level, the approach to computing the model of $\mathcal{K}$ follows that of computing the stable model of a logic program $\mathcal{P}$ alone. To compute a stand-alone stable model, $\mathcal{P}$ is grounded, an LP-interpretation J is guessed, and J is checked to determine whether it is a subset-minimal model of the reduct of $\mathcal{P}$ relative to J (cf. Section 2.2). To extend this approach to an MKNF knowledge base $\mathcal{K}$, $\mathsf{ground}(\mathcal{K})$ is first constructed using the constants in $\mathfrak{C}$, and a set $J \subseteq \mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ is guessed.[10] Then using Theorem 14, it is checked whether J corresponds to an MKNF model by determining whether $\mathsf{OB}_{\mathcal{O},J}$ entails all atoms in J, does not entail head atoms in $\mathsf{HA}(\mathsf{ground}(\mathcal{P})) \setminus J$, and that J is a subset-minimal model. It was shown in [77] that the models of $\mathcal{K}$ coincide with those where $\mathcal{K}$ is grounded in the above manner.

Clearly, computing the model of $\mathcal{P}$ will be decidable if $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ is finite, and if $\mathsf{OB}_{\mathcal{O},J}$ is decidable for all $J \subseteq \mathsf{HA}(\mathsf{ground}(\mathcal{P}))$. When $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ is finite, the latter condition will hold if the underlying DL of $\mathcal{O}$ is decidable, and if $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ contains only Abox assertions or sentences that correspond to axioms in the underlying description logic of $\mathcal{O}$. The condition of *DL-safety* is sufficient to guarantee that $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ is finite, and is informally explained as follows (see Theorem 6.2 of [77] for technical details). A *non-DL atom* is an atom whose predicate symbol only occurs in $\mathcal{P}$. (The predicate $\approx$, however is always considered a DL-atom.) A rule π is DL-safe if each variable in π occurs in a positive non-DL atom in the body of π; a program is DL-safe if all of its rules are. Clearly DL-safety, which is defined using the finite set of predicate symbols in an MKNF knowledge base, is a decidable condition for finiteness of $\mathcal{P}$; and when entailment by $\mathcal{O}$ is also decidable, DL-safety is a sufficient condition for entailment by $\mathcal{K}$ itself.

[9] Due to the faithfulness of the MKNF semantics with respect to the underlying program and ontology of a knowledge base, we abuse notation slightly and use $\models$ for entailment by first-order theories, logic programs, and MKNF knowledge bases depending on the context.

[10] Note that $\mathsf{HA}(\mathsf{ground}(\mathcal{P}))$ is not an LP-interpretation as it may contain ground generalised atoms.

To efficiently implement MKNF reasoners, both the reasoners for $\mathcal{O}$ and $\mathcal{P}$ must be able to call one another at a fine-grained level. We know of no MKNF reasoners that support the full stable model semantics for $\mathcal{P}$. However, in the important special case where $\mathcal{P}$ is a stratified program, MKNF reasoners have been implemented based on tabled resolution for MKNF [8].[11] These include a stand-alone reasoner where the underlying *DL* of $\mathcal{O}$ is based on $\mathcal{ALCQ}$ [44], and the NoHR plug-in for the Protégé ontology editor where the underlying *DL* of $\mathcal{O}$ is based on either $\mathcal{EL}$ [53] or *DL-Lite* [23].

2.5. Ontology updates

An *update* is typically described as an operation that brings a knowledge base *up to date* when the *world described by it changes* [56,57,96]. From a generic perspective, update operators were studied within the context of propositional logic. We present the relevant concepts in the next subsection and subsequently discuss their first-order generalisations that are used to perform updates of DL ontologies [15,27–29,36,58,73].

Belief update postulates and operators Propositional update operators are formalised as functions that take two propositional formulas, representing the original knowledge base and its update, as arguments, and return a formula representing the updated knowledge base. Any such operator $\diamond$ is inductively generalised to finite sequences of propositional formulas $\langle \phi_i \rangle_{i<n}$ as follows: $\diamond\langle\phi_0\rangle = \phi_0$ and $\diamond\langle\phi_i\rangle_{i<n+1} = (\diamond\langle\phi_i\rangle_{i<n}) \diamond \phi_n$. To further specify the desired properties of update operators, the following eight postulates for a belief update operator $\diamond$ and formulas ϕ, ψ, μ, ν were proposed [56]:

(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 has a single model, 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* [24]. Sometimes it also becomes useful to consider weakenings of these postulates, such as the following ones [51]:

(B2.$\top$) $\phi \diamond \top \equiv \phi$.
(B4.1) If $\phi \equiv \psi$, then $\phi \diamond \mu \equiv \psi \diamond \mu$.
(B4.2) If $\mu \equiv \nu$, then $\phi \diamond \mu \equiv \phi \diamond \nu$.
(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 reader can verify, (B2.$\top$) is an immediate consequence of (B2) while (B4.1) and (B4.2) are together equivalent to (B4). Furthermore, in the presence of (B4.1), it can be shown that (B8.1) and (B8.2) are together equivalent to (B8).

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, an important representation theorem makes it possible to constructively characterise and evaluate every operator $\diamond$ that satisfies postulates (B1)–(B8) [56]:

Theorem 16 *(Belief update representation theorem). (See [56].) A belief update operator $\diamond$ satisfies conditions (B1)–(B8) if and only if there exist strict preorders $<^I$ for each interpretation I such that $I <^I J$ for all interpretations $J \neq I$, and for all formulas ϕ, μ,*

$$[\![\phi \diamond \mu]\!] = \bigcup_{I \in [\![\phi]\!]} \left\{ J \in [\![\mu]\!] \;\middle|\; \neg\exists K \in [\![\mu]\!] : K <^I J \right\} . \tag{5}$$

In words, an updated formula has models $[\![\phi \diamond \mu]\!]$ that are models of the update μ and that are minimal for the preorder $<^I$ for some model of the original formula ϕ.[12] Katsuno and Mendelzon's results provide a framework for belief update operators, each specified on the semantic level by strict preorders assigned to each propositional interpretation. The most influential instance of this framework is the *Possible Models Approach* by [95], based on minimising the set of atoms

[11] For non-stratified programs, the well-founded semantics for MKNF is adopted [59].
[12] A *preorder* is a reflexive and transitive binary relation; a *strict* preorder is an irreflexive and transitive binary relation.

whose truth value changes when an interpretation is updated. Formally, for all interpretations I, J and K, the strict preorder $<^I_{\mathsf{W}}$ is defined as follows: $J <^I_{\mathsf{W}} K$ if and only if $(J \div I) \subsetneq (K \div I)$, where $\div$ denotes set-theoretic symmetric difference. The operator $\diamond_{\mathsf{W}}$ by Winslett, unique up to equivalence of its inputs and output, thus satisfies the following equation:

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

Note that it follows from Theorem 16 that $\diamond_{\mathsf{W}}$ satisfies postulates (B1)–(B8).

First-order update postulates and operators The preceding approach to belief update can, for the most part, be generalised from propositional theories to deal with updates of first-order theories. In this context, a *first-order update operator* is a binary function on the set of all first-order theories and any such operator $\diamond$ is inductively generalised to finite sequences of first-order theories $\langle \mathcal{T}_i \rangle_{i<n}$ as follows: $\diamond\langle \mathcal{T}_0 \rangle = \mathcal{T}_0$ and $\diamond\langle \mathcal{T}_i \rangle_{i<n+1} = (\diamond\langle \mathcal{T}_i \rangle_{i<n}) \diamond \mathcal{T}_n$. A single first-order sentence ϕ is updated by treating it as the theory $\{ \phi \}$. Furthermore, the update postulates discussed previously, except for (B7) and (B8.1) which require disjunction of a pair of theories to be defined, can be directly generalised to the first-order case:

(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.$\top$) $\mathcal{T} \diamond \emptyset \equiv \mathcal{T}$.
(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 $\mathcal{T} \diamond \mathcal{U} \models \mathcal{S} \diamond \mathcal{U}$.

Whereas *postulates* are meant to encode properties that are assumed or claimed to be true, their role in this paper is distinct. We are not claiming that first-order update operators should obey all of these. They are simply taken as *properties* that can serve to examine different classes of first-order update operators. For instance, in Section 3 we prove results that hold for all first-order update operators that satisfy (FO8.2). Nevertheless, because they were lifted from the belief update *postulates* for the propositional case, we decided to call them *postulates*.

Winslett's operator for first-order theories [96] is defined just as its propositional counterpart, using equation (5), the difference is only that the strict preorders $<^I_{\mathsf{W}}$ are defined on predicate symbols in $\mathcal{P}$ (Section 2.1) and for all interpretations $I, J, K \in \mathcal{I}_{\mathcal{L}}$ as follows. $J <^I_{\mathsf{W}} K$ if and only if

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

Winslett's operator also forms the formal basis for ontology updates [15,27–29,36,58,73]. The basic idea here is quite simple: assuming that $\diamond$ is some first-order update operator, an update of an ontology $\mathcal{O}_1$ by an ontology $\mathcal{O}_2$ is determined by the first-order theory $\kappa(\mathcal{O}_1) \diamond \kappa(\mathcal{O}_2)$. However, it may easily happen that this theory is not expressible in the Description Logic used to encode $\mathcal{O}_1$ and $\mathcal{O}_2$ [10]. This problem has been addressed in the literature, either by identifying Description Logics that are closed with respect to applications of Winslett's operator, by using approximation techniques to arrive at the closest ontology that is still expressible in the original Description Logic, or finding a more expressive Description Logic in which the updates can be expressed. Whereas this is a very challenging, mostly open problem, it falls outside the focus point of this paper. So, in order to be able to proceed, and focus on issues with *hybrid updates*, in this paper we abstract away from these as well as the related representational issues, and will assume that they have been addressed in a suitable manner.

2.6. Rule updates

State-of-the-art rule update semantics are based on fundamentally different principles and methods when compared to their ontology update counterparts. In the following we define one of the most advanced rule update semantics, the *refined dynamic stable models* for sequences of logic programs [7].

A rule update semantics provides a way to assign stable models to a pair or sequence of programs where each component of the pair or sequence represents an update of the preceding ones. Formally, a *dynamic logic program* (DLP) is a finite sequence of logic programs and by $\mathsf{all}(\boldsymbol{P})$ we denote the multiset of all rules in the components of $\boldsymbol{P}$. A rule update semantics S assigns a *set of* S*-models*, denoted by $[\![\boldsymbol{P}]\!]_{\mathsf{S}}$, to $\boldsymbol{P}$.

We concentrate on semantics based on the causal rejection principle [6,7,13,37,65,66,78] which states that a rule is *rejected* if it is in a direct conflict with a more recent rule. The basic type of conflict between rules π and σ 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 [65], 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 $\boldsymbol{P}$ contains a rule with an objective literal l in its head, its expansion $\boldsymbol{P}^{\mathsf{e}}$ 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:

$$\mathcal{P}_i^{\mathsf{e}} = \mathcal{P}_i \cup \left\{ \sim\bar{l} \leftarrow \mathsf{B}_\pi. \;\middle|\; \pi \in \mathcal{P}_i \land \mathsf{H}_\pi = l \land l \in \mathit{Lits}_\mathsf{G} \right\} .$$

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 conflicting 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 $\mathsf{H}_\pi = \sim\mathsf{H}_\sigma$.

Based on such conflicts and on a stable model candidate, a *set of rejected rules* can be determined and it can be verified that the candidate is indeed stable w.r.t. the remaining rules. One of the most mature of these semantics, the *refined dynamic stable models* [7], or RD-*semantics*, is defined using a fixed point equation. Given a DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ and an interpretation J, the multisets of rejected rules $\mathsf{rej}(\boldsymbol{P}, J)$ and of default assumptions $\mathsf{def}(\boldsymbol{P}, J)$ are defined as follows:

$$\mathsf{rej}(\boldsymbol{P}, J) = \left\{ \pi \in \mathcal{P}_i \;\middle|\; i < n \land \exists j \geq i\, \exists \sigma \in \mathcal{P}_j : \pi \bowtie \sigma \land J \models \mathsf{B}_\sigma \right\} ,$$
$$\mathsf{def}(\boldsymbol{P}, J) = \{ \sim l \mid l \in \mathit{Lits}_\mathsf{G} \land \neg(\exists \pi \in \mathsf{all}(\boldsymbol{P}) : \mathsf{H}_\pi = l \land J \models \mathsf{B}_\pi) \} .$$

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

$$J^* = \mathsf{least}\left([\mathsf{all}(\boldsymbol{P}^{\mathsf{e}}) \setminus \mathsf{rej}(\boldsymbol{P}^{\mathsf{e}}, J)] \cup \mathsf{def}(\boldsymbol{P}^{\mathsf{e}}, J)\right) \tag{6}$$

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

Example 17. (See [66].) Consider an agent with beliefs represented by the following program $\mathcal{P}_0$:

$$\mathsf{GoHome} \leftarrow \sim\mathsf{Money}. \tag{7}$$

$$\mathsf{GoRestaurant} \leftarrow \mathsf{Money}. \tag{8}$$

$$\mathsf{Money}. \tag{9}$$

The only stable model of $\mathcal{P}_0$ is $J' = \{ \mathsf{Money}, \mathsf{GoRestaurant} \}$, capturing that the agent has money by rule (9), so according to rule (8) it plans to go to a restaurant. Now consider an update $\mathcal{P}_1$ with the following two rules:

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

The intended result is that, after the update by $\mathcal{P}_1$, GoRestaurant be false because its only justification, Money, is no longer true. Furthermore, we expect GoHome to become true, i.e. the rule (7) should be triggered because Money became false. With $J = \{ \mathsf{Robbed}, \mathsf{GoHome} \}$, we have $\mathsf{rej}(\langle \mathcal{P}_0, \mathcal{P}_1 \rangle^{\mathsf{e}}, J) = \{ \mathsf{Money}. \}$, and J obeys the stability condition (6). Furthermore, J is the only interpretation with these properties.

The following more abstract example aims at illustrating, with greater detail, the concepts of rejected rules $\mathsf{rej}(\boldsymbol{P}, J)$, default assumptions $\mathsf{def}(\boldsymbol{P}, J)$, and the stability condition (6).

Example 18. Consider the logic programs $\mathcal{P}_0$, $\mathcal{P}_1$, $\mathcal{P}_2$ with

$$\begin{array}{llllll}
\mathcal{P}_0: & p \leftarrow s. & \mathcal{P}_1: & \sim s \leftarrow u. & \mathcal{P}_2: & \sim u. \\
& q \leftarrow \sim s. & & t \leftarrow \sim s. & & \neg s. \\
& r \leftarrow \neg s, t. & & u \leftarrow \sim r. & & \\
& s. & & & &
\end{array}$$

Their expanded versions are $\mathcal{P}_0^{\mathsf{e}}$, $\mathcal{P}_1^{\mathsf{e}}$, $\mathcal{P}_2^{\mathsf{e}}$ with

$$\begin{array}{lllllllll}
\mathcal{P}_0^{\mathsf{e}}: & p \leftarrow s. & \sim\neg p \leftarrow s. & \mathcal{P}_1^{\mathsf{e}}: & \sim s \leftarrow u. & & \mathcal{P}_2^{\mathsf{e}}: & \sim u. & \\
& q \leftarrow \sim s. & \sim\neg q \leftarrow \sim s. & & t \leftarrow \sim s. & \sim\neg t \leftarrow \sim s. & & \neg s. & \sim s. \\
& r \leftarrow \neg s, t. & \sim\neg r \leftarrow \neg s, t. & & u \leftarrow \sim r. & \sim\neg u. \leftarrow \sim r. & & & \\
& s. & \sim\neg s. & & & & & &
\end{array}$$

The interpretation $J = \{ q, t, u \}$ is the only dynamic stable model of the DLP $\langle \mathcal{P}_0, \mathcal{P}_1 \rangle$. To check, we first determine that $\mathsf{rej}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}} \rangle, J) = \{ s. \}$, $\mathsf{def}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}} \rangle, J) = \sim\{ p, r, \neg p, \neg q, \neg r, \neg s, \neg t, \neg u \}$, and $J^* = \{ q, t, u \} \cup \sim\{ p, r, s, \neg p, \neg q, \neg r, \neg s, \neg t, \neg u \}$. Then, we can confirm that

$$J^* = \mathsf{least}\left([\mathsf{all}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}} \rangle) \setminus \mathsf{rej}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}} \rangle, J)] \cup \mathsf{def}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}} \rangle, J) \right)$$

with all literals treated as atoms. If we now consider the DLP consisting of the three programs, $\langle \mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2 \rangle$, the interpretation $J = \{ q, r, \neg s, t \}$ is its only dynamic stable model. The sets of rejected rules and defaults are $\mathsf{rej}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}}, \mathcal{P}_2^{\mathsf{e}} \rangle, J) = \{ s., \sim\neg s., u \leftarrow \sim r. \}$, $\mathsf{def}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}}, \mathcal{P}_2^{\mathsf{e}} \rangle, J) = \sim\{ p, u, \neg p, \neg q, \neg r, \neg t, \neg u \}$, and $J^* = \{ q, r, \neg s, t \} \cup \sim\{ p, s, u, \neg p, \neg q, \neg r, \neg t, \neg u \}$. Then, we can confirm that

$$J^* = \mathsf{least}\left([\mathsf{all}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}}, \mathcal{P}_2^{\mathsf{e}} \rangle) \setminus \mathsf{rej}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}}, \mathcal{P}_2^{\mathsf{e}} \rangle, J)] \cup \mathsf{def}(\langle \mathcal{P}_0^{\mathsf{e}}, \mathcal{P}_1^{\mathsf{e}}, \mathcal{P}_2^{\mathsf{e}} \rangle, J) \right)$$

again, with all literals treated as atoms.

3. Dynamic MKNF knowledge bases with static rules

Now that all necessary technical background has been covered, we are ready to start addressing the problem of updating hybrid knowledge bases. We adopt MKNF knowledge bases (cf. Section 2.4) as the foundation for representing hybrid knowledge and, as with the rule update semantics, our general goal is to assign *dynamic MKNF models* to sequences of MKNF knowledge bases where each component represents an update of the preceding ones.[13]

We thus introduce the following concept:

Definition 19 *(Dynamic MKNF knowledge base).* A finite sequence of MKNF knowledge bases is called a *dynamic MKNF knowledge base* (or *DMKB* for short).

In this section we address a restricted 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. Even under this assumption, knowledge bases allow for reasoning with assumptions and for exceptions to be naturally expressed, yet still support 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. This operator is augmented with a first-order update operator that deals with ontology updates and is then used to define dynamic MKNF models of DMKBs with static rules. The semantics is thus *parameterised* by a first-order update operator; we particularly consider Winslett's operator as one way to instantiate the semantics and show how it can handle updates of the hybrid knowledge base presented in Example 1.

As discussed in Section 2.1, we assume a language $\mathcal{L}$, and, without loss of generality, restrict our attention to SNA interpretations over $\mathcal{L}$ (Definition 3). In addition, two important assumptions are made for the development of this section.

- As in most existing work on rule updates, we assume that all MKNF rules are ground: in our case, this is done with respect to $\mathcal{C}$, the set of constants in $\mathcal{L}$, as discussed in Section 2.4.
- We consider rules without default negation in their heads: as shown in Proposition 15, this restriction does not remove any expressivity. However, we do allow rules to contain generalised atoms as introduced in Section 2.4.

In the remainder of this section, we start by defining the consequence operator for characterising MKNF models of MKNF knowledge bases (Section 3.1), then proceed by imbuing it with the ability to perform ontology updates and using it to define an update semantics for DMKBs with static rules (Section 3.2), for those update operators that satisfy the update postulate (FO8.2). Finally, we establish the basic formal properties of this hybrid update semantics (Section 3.3). The relevant proofs can be found in Appendix B.[14]

---

[13] Dealing with sequences of MKNF knowledge bases instead of just updating an MKNF knowledge base by another MKNF knowledge base, while relying on a binary update operator which works on pairs, allows us to abstract away from representation difficulties that already arise when updating DLs and LPs alone. In order to work with pairs and still be able to allow for more than one update, the result of the update operation must be representable in a way that is acceptable to the operator. While this is the case for some (though not most) DL operators, it is not the case for most LP update operators. In fact, when it comes to LP update operators, the result of updating a logic program by another logic program is not, in general, a logic program of the same class, written with the same alphabet. Indeed, it is known to be theoretically impossible to have the result of such an update be one logic program of the same class, while still obeying some very basic properties of updates [65,89].

[14] Preliminary versions of this work have been published in [83,84]. In contrast to 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. A static consequence operator and its fixed points

To define our consequence operator and its fixed points, we first consider MKNF knowledge bases with positive programs, and then extend our approach to programs with default negation.

3.1.1. Positive programs

The consequence operator for MKNF knowledge bases is an extension to the usual two-valued immediate consequence operator from Logic Programming (cf. e.g., [74]): given a positive MKNF program (i.e., Section 2.4), the operator returns the heads of all MKNF rules whose bodies are satisfied by the argument interpretation. This idea can be generalised to deal with *positive MKNF knowledge bases* (i.e., MKNF knowledge bases whose underlying programs are positive) by conjoining the heads of rules whose bodies are satisfied together with the translation of the ontology, and returning the set of models of the resulting first-order theory. Formally:

Definition 20 *(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[15]:

$$T_\mathcal{P}(\mathcal{M}) = \{ \mathsf{H}_\pi \mid \pi \in \mathcal{P} \wedge \mathcal{M} \models \kappa(\mathsf{B}_\pi) \} \ ,$$

$$T_\mathcal{K}(\mathcal{M}) = [\![T_\mathcal{P}(\mathcal{M}) \cup \kappa(\mathcal{O})]\!] \ .$$

An analogous consequence operator on sets of modal atoms was independently presented in [77]. Note also that there are important differences between the consequence operator of Definition 20 and the usual consequence operator in Logic Programming. As the set of **K**-atoms inferred from a positive MKNF knowledge base is anti-monotonic with respect to elements of $\mathfrak{M}$, the $T_\mathcal{P}$ operator, which maps an MKNF interpretation $\mathcal{M}$ to the **K** atoms that can be inferred under the MKNF semantics, will be anti-monotonic. The $T_\mathcal{K}$ operator on the other hand, maps an MKNF interpretation $\mathcal{M}$ to the greatest MKNF interpretation that satisfies $T_\mathcal{P}(\mathcal{M}) \cup \kappa(\mathcal{O})$ and so is monotonic. Implicit in the use of $T_\mathcal{K}$, is the representation from Definition 9 of an MKNF interpretations as a set of first-order interpretations $\mathcal{M} \subseteq \mathfrak{I}_\mathcal{L}$ such that for an MKNF sentence ϕ, $\mathcal{M} \models \phi$ iff $(I, \mathcal{M}, \mathcal{M}) \models \phi$ for all $I \in M$.[16] With this in mind, we view the set $\mathfrak{M}$ of all MKNF interpretations (including the empty set) as a complete lattice under set inclusion with the greatest element $\mathfrak{I}_\mathcal{L}$.

Proposition 21 *(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 72. □

In order to make use of $T_\mathcal{K}$, we construct its greatest fixed point on the lattice of MKNF interpretations, and show in Propositions 23 and 26 that any MKNF model corresponds to this greatest fixed point. Note that construction of a greatest fixed point is adopted for notational ease, since larger MKNF interpretations reflect less knowledge (via the **K** operator) than MKNF interpretations.[17] The following example illustrates how $T_\mathcal{K}$ is iterated, starting from the greatest element $\mathfrak{I}_\mathcal{L}$, until a fixed point is reached which coincides with the MKNF model of the MKNF knowledge base.

Example 22 *(Iterating the consequence operators).* Consider the MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ where[18]

$$\mathcal{O}: \ p \vee \neg q \qquad \mathcal{P}: \ \begin{array}{l} q \leftarrow r. \\ q \leftarrow s. \\ r. \end{array}$$

Starting from the interpretation $\mathcal{M}_0 = \mathfrak{I}_\mathcal{L}$, 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:

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

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

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

[15] Recall from Definition 9 that $\mathfrak{M}$ denotes the set of all MKNF interpretations including the empty set.

[16] The justification for the use of such representations is presented in Appendix B, Lemma 106.

[17] Of course, for computational purposes, least fixed points are used to compute actual MKNF models [59,77].

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

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

Since $T_{\mathcal{K}}$ is monotonic (Proposition 21), 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 23 *(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 73. □

3.1.2. Programs with default negation

To add support for default negation in bodies of MKNF rules, we use essentially the same strategy as the one used for defining stable models [42]. 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 whose negative bodies are satisfied in $\mathcal{M}$.

Definition 24 *(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

$$\mathcal{P}^{\mathcal{M}} = \left\{ \mathsf{H}_\pi \leftarrow \mathsf{B}_\pi^+. \mid \pi \in \mathcal{P} \land \mathcal{M} \models \kappa\left(\sim\mathsf{B}_\pi^-\right) \right\} .$$

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 25 *(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 22 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 22. 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 26 *(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 74. □

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 $T_{\mathcal{K}}$, similar to 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. At the same time, we constrain ourselves to dealing with are DMKBs that have *static rules*, as defined here:

Definition 27 *(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 $0 < i < n$.

3.2.1. Updating positive DMKBs with static rules

As a first step, assume that we are given a DMKB with static rules $\boldsymbol{K}$ and also 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 28 *(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:

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

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

Example 29 *(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:

$$\begin{array}{llllll} \mathcal{O}_0: & p \vee \neg q & \mathcal{P}_0: & q \leftarrow r. & \mathcal{O}_1: & \neg r \wedge \neg s \\ & & & q \leftarrow s. & & \\ & & & r. & & \\ & & & s. & & \end{array}$$

Clearly, $\boldsymbol{K}$ has static rules. We can thus iteratively apply the updating consequence operator (parameterised by the Winslett update operator of Section 2.5), $T^{\diamond_W}_{\boldsymbol{K}}$, starting from the MKNF interpretation $\mathcal{M}_0 = \mathfrak{I}_{\mathcal{L}}$, until we reach a fixed point. After the first application we obtain the following:

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

Furthermore, $T^{\diamond_W}_{\boldsymbol{K}}(\mathcal{M}_1) = \mathcal{M}_1$ 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 $T^{\diamond_W}_{\boldsymbol{K}}$ 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 30 *(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 74. □

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 31 *(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.

3.2.2. Updating non-positive DMKBs with static rules

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 32 *(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 33 *(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 25, namely $\mathcal{O} = \{ p \lor \neg q \}$ and $\mathcal{P}'$ consists of the clauses $\{ q \leftarrow r, q \leftarrow s, r \leftarrow \sim s, s \leftarrow \sim r \}$. $\mathcal{K}_1$ is as in Example 29, i.e., $\mathcal{K}_1 = (\mathcal{O}_1, \emptyset)$, where $\mathcal{O}_1 = \{ \neg r \land \neg s \}$.

Consider the MKNF interpretation $\mathcal{M}_1 = [\![\{ p \lor \neg q, \neg r, \neg s \}]\!]$. 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 29: $\{ q \leftarrow r, q \leftarrow s, r, s \}$. This is 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 $\diamond_W$-dynamic MKNF model of $\boldsymbol{K}$, we declare it to be the *$\diamond_W$-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 $\mathcal{K}_1$.

Now we can 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 34 *(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 consequence relation by considering the credulous consequences of $\diamond$-dynamic MKNF models.

Definition 35 *(Consequence relation).* Let $\diamond$ be a first-order update operator that satisfies (FO8.2), $\boldsymbol{K}$ a DMKB with static rules and $\mathcal{T}$ an MKNF theory. We say that $\boldsymbol{K}$ *$\diamond$-MKNF entails $\mathcal{T}$*, denoted by $\boldsymbol{K} \models^{\diamond}_{\mathsf{MKNF}} \mathcal{T}$, if $\mathcal{M} \models \mathcal{T}$ for some $\diamond$-dynamic MKNF model $\mathcal{M}$ of $\boldsymbol{K}$.

Credulous and other types of logical consequence can be obtained analogously.

Let us now demonstrate the defined update semantics on a simple example in which we use Winslett's first-order operator $\diamond_W$ for performing ontology updates:

Example 36 *(Updating the electronic marketplace knowledge base).* Consider again the simple hybrid knowledge base about an electronic marketplace presented in Example 1, formalised as the MKNF knowledge base $\mathcal{K}_0 = (\mathcal{T}, \mathcal{P})$ where

$$\mathcal{T}: \quad \mathsf{Seller} \sqsubseteq \mathsf{User}$$
$$\mathsf{ProspectiveSeller} \equiv \neg\mathsf{Seller} \sqcap \exists\mathsf{RecommendedBy}.\mathsf{Seller}$$
$$\mathcal{P}: \quad \neg\mathsf{User}(\mathbf{x}) \leftarrow \sim\mathsf{User}(\mathbf{x}).$$
$$\mathsf{PaysServiceFee}(\mathbf{x}) \leftarrow \mathsf{Seller}(\mathbf{x}), \sim\mathsf{Student}(\mathbf{x}).$$

Throughout this example, we will be keeping the terminological assertions in the TBox $\mathcal{T}$ static by reasserting them in each update. We also assume that the constant symbol *john* is part of the language.

Initially, all we can conclude from the DMKB $\boldsymbol{K}_0 = \langle \mathcal{K}_0 \rangle$ is that *john* does not belong to the concept User because of the first rule in $\mathcal{P}$, and, consequently, he cannot belong to the concept Seller because of the first axiom in $\mathcal{T}$. Formally:

$$\boldsymbol{K}_0 \models^{\diamond_W}_{\mathsf{MKNF}} \{ \neg\mathsf{User}(john), \neg\mathsf{Seller}(john) \} \ .$$

Now consider an update by the ABox $\mathcal{A}_1 = \{ \exists\mathsf{RecommendedBy}.\mathsf{Seller}(john) \}$, expressing that a new recommendation for *john* has been made by some member of the concept Seller. In order to perform this update, we form the DMKB $\boldsymbol{K}_1 = \langle \mathcal{K}_0, \mathcal{K}_1 \rangle$ where $\mathcal{K}_1 = (\mathcal{A}_1 \cup \mathcal{T}, \emptyset)$. We conclude the following:

$$\boldsymbol{K}_1 \models^{\diamond_W}_{\mathsf{MKNF}} \{ \neg\mathsf{User}(john), \neg\mathsf{Seller}(john),$$
$$\exists\mathsf{RecommendedBy}.\mathsf{Seller}(john), \mathsf{ProspectiveSeller}(john) \} \ .$$

In other words, we can now additionally conclude that the assertion in $\mathcal{A}_1$ is true and, due to the second axiom in $\mathcal{T}$, *john* is now a ProspectiveSeller.

Based on this information, *john* is offered, and agrees, to become a Seller at the marketplace. This update is formalised as the ABox $\mathcal{A}_2 = \{ \mathsf{Seller}(john) \}$, resulting in the DMKB $\boldsymbol{K}_2 = \langle \mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2 \rangle$ where $\mathcal{K}_2 = (\mathcal{A}_2 \cup \mathcal{T}, \emptyset)$, which allows us to conclude the following:

$$\boldsymbol{K}_2 \models^{\diamond\mathsf{w}}_{\mathsf{MKNF}} \{ \ \mathsf{User}(john), \mathsf{Seller}(john), \exists\mathsf{RecommendedBy}.\mathsf{Seller}(john), \mathsf{PaysServiceFee}(john) \} \ .$$

Note in particular that *john* is now both a Seller and a User, due to the first axiom in $\mathcal{T}$, and that he is no longer a ProspectiveSeller because of its definition in $\mathcal{T}$ and because he became a Seller.

Subsequently, *john* makes an inquiry to not be charged the service fee until his business becomes more established, and his request is approved. This update is represented as the ABox $\mathcal{A}_3 = \{ \ \neg\mathsf{PaysServiceFee}(john) \ \}$, resulting in the DMKB $\boldsymbol{K}_3 = \langle \mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3 \rangle$ where $\mathcal{K}_3 = (\mathcal{A}_3 \cup \mathcal{T}, \emptyset)$ and leading to the following consequences:

$$\boldsymbol{K}_3 \models^{\diamond\mathsf{w}}_{\mathsf{MKNF}} \{ \ \mathsf{User}(john), \mathsf{Seller}(john), \exists\mathsf{RecommendedBy}.\mathsf{Seller}(john), \neg\mathsf{PaysServiceFee}(john) \} \ .$$

Note in particular that though the body of the second rule in $\mathcal{P}$ is satisfied, its head is automatically overridden by the ontology update.

Furthermore, it is later discovered that *john* is under eighteen years old, which is against the rules of the marketplace. His user account becomes disabled and any recommendations deleted, as expressed by the ABox $\mathcal{A}_4 = \{ \ \neg\mathsf{User}(john), \neg\exists\mathsf{RecommendedBy}.\top(john) \ \}$. The consequences of the DMKB $\boldsymbol{K}_4 = \langle \mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4 \rangle$, where $\mathcal{K}_4 = (\mathcal{A}_4 \cup \mathcal{T}, \emptyset)$, are then as follows:

$$\boldsymbol{K}_4 \models^{\diamond\mathsf{w}}_{\mathsf{MKNF}} \{ \ \neg\mathsf{User}(john), \neg\mathsf{Seller}(john), \neg\exists\mathsf{RecommendedBy}.\top(john), \neg\mathsf{PaysServiceFee}(john) \} \ .$$

3.3. Properties and relations

In the following we look more closely at the 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 37 *(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 74. □

Note that a consequence of this result and of Proposition 15 (Section 2.4) is that the introduced update semantics is also faithful w.r.t. the ontology semantics and the stable model semantics.

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 38 *(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 75. □

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* [24], 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 39 *(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 75. □

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 this desirable property is notoriously problematic when rule updates are considered, as witnessed by the many rule update semantics that violate it [6,30,31,37,65,66,80,97].

Theorem 40 *(Immunity to tautological updates). Suppose that $\diamond$ satisfies (FO2.$\top$) 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*

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

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

Proof. See Appendix B, page 75. □

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 does not affect the resulting dynamic MKNF models. 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 41 *(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 75. □

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

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

have the same stable models and are even strongly equivalent [72], 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 $\diamond_W$-dynamic MKNF model $\mathcal{M}$ such that $\mathcal{M} \models p$ while $\langle (\emptyset, \mathcal{Q}), (\mathcal{O}, \emptyset) \rangle$ has a $\diamond_W$-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 [65], this property is satisfied.

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:

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

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

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

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 $\langle (\emptyset, \mathcal{P}), (\mathcal{O}', \emptyset) \rangle \models^{\diamond_W}_{\mathsf{MKNF}} (\emptyset, \mathcal{P})$ because $\langle (\emptyset, \mathcal{P}), (\mathcal{O}', \emptyset) \rangle$ has two $\diamond_W$-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* [34,75]. Hence it is likely that (FO2) is not going to be satisfied by any hybrid update semantics that is faithful to the stable model semantics.

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

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

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 $\diamond_W$-dynamic MKNF model.[19]

[19] Whereas the absence of a $\diamond_W$-dynamic MKNF model may appear unexpected, it reflects the way logic programming rules are handled when subject to updates, namely the requirement that a concrete cause for rejection exists (other than simply the fact that the update results in an inconsistency). As a concrete example, suppose a logic program encodes the well-known n-queen problem, and the ontology – subject to updates – encodes the queens that are already placed on the chessboard. Models will correspond to possible solutions to the n-queen problem given the already placed queens. To enforce this, the program would have a rule, acting as an integrity constraint, guaranteeing that no two queens can be placed in the same column. This rule would have a form similar to that used in the counter example i.e. $p \leftarrow q, \sim p.$, where q encodes the fact that two queens are placed in the same column. If we update the ontology to add two queens to the same column, making q true in the ontology, the expected result is that we obtain no $\diamond_W$-dynamic MKNF models, encoding the fact that there is no solution with the queens in the update.

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.

4. Splitting properties

In this section we prepare the formal groundwork for introducing a hybrid update semantics to complement the semantics defined in Section 3 – instead of requiring that the rule part of a hybrid knowledge base remain static, it makes both ontology and rule updates possible at the cost of placing certain constraints on the interaction between the ontology and rules. These constraints are inspired by the ideas behind splitting theorems for Logic Programs as formulated in [71] on which we focus our attention here. Splitting properties have also been studied for various generalizations of stable models, and form the basis for several approaches to defining modules of logic programs under the stable model semantics [12,54].

First we formulate the splitting properties in an abstract manner, so they can be easily instantiated for any semantic framework (Section 4.1). Then we show that prominent static hybrid semantics as well as ontology and rule update semantics satisfy these properties (Section 4.2).

4.1. Abstract splitting properties

Splitting properties were first studied by [71] 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. For instance, 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 the subprograms is a stable model of $\mathcal{P}$.

The splitting properties of [71] take this idea further by allowing subprograms to share literals in a constrained, cascading manner. Assume that we aim for a split 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 section is to prepare the formal groundwork for defining 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 5 where we also show that if both the first-order update operator and rule update semantics have the *Abstract Splitting Properties* and satisfy two basic properties of updates, 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 different formalisms – not just normal logic programs under the stable model semantics [71], but also default logic, as was done in [92], and MKNF knowledge bases, first-order update operators, rule update and hybrid update semantics, as we do in Section 4.2. Some of the abstractions we use for this purpose, such as those for logical formalisms and semantics, are inspired by similar abstractions in [17].

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

Definition 42 *(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 LP interpretations, then $I \uplus J = I \cup J$ and if $I \cup J$ contains a pair of complementary literals, then it is itself not an LP 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 the case of LP 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 $\mathcal{T} \in \mathfrak{T}$, for every splitting set U for $\mathcal{T}$, and for 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 only of U, 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 43 *(Splitting set, bottom, top and reduct of logic programs).* For logic programs under the stable model semantics, $\mathfrak{T}$ is the set of all logic programs, $\mathfrak{S}$ is the set of all LP interpretations and every splitting set is some set of objective literals [71]. 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 LP 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 subsection 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 to define what it means for a semantics S to *satisfy the Abstract Splitting Set Property*. Assuming that the notions of splitting set, bottom, top and reduct are known for the logical formalism $(\mathfrak{T}, \mathfrak{S})$, we can define a *solution w.r.t. a splitting set* as follows:

Definition 44 *(Solution w.r.t. a splitting set).* Let S be a semantics for $(\mathfrak{T}, \mathfrak{S})$ and U a splitting set for a theory $\mathcal{T} \in \mathfrak{T}$. An S-*solution to* $\mathcal{T}$ *w.r.t.* U is a pair of semantic structures $(\mathcal{X}, \mathcal{Y})$ such that $\mathcal{X} \in [\![b_U(\mathcal{T})]\!]_{\mathsf{S}}$ and $\mathcal{Y} \in [\![e_U(\mathcal{T}, \mathcal{X})]\!]_{\mathsf{S}}$ and $\mathcal{X} \uplus \mathcal{Y} \in \mathfrak{S}$.

The Abstract 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 45 *(Abstract splitting set property).* We say that a semantics S for $(\mathfrak{T}, \mathfrak{S})$ *satisfies the Abstract Splitting Set Property* if for all theories $\mathcal{T} \in \mathfrak{T}$ for which $[\![\mathcal{T}]\!]_{\mathsf{S}}$ is defined and every splitting set U for $\mathcal{T}$,

$$[\![\mathcal{T}]\!]_{\mathsf{S}} = \{\, \mathcal{X} \uplus \mathcal{Y} \mid (\mathcal{X}, \mathcal{Y}) \text{ is an } \mathsf{S}\text{-solution to } \mathcal{T} \text{ w.r.t. } U \,\} \quad .$$

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 Abstract Splitting Set Property. This part of the theory relies on transfinite sequences of sets, so we first introduce the following basic concepts:

Definition 46 *(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 $\mathfrak{U}$, we define a *splitting sequence* as follows:

Definition 47 *(Splitting sequence).* A *splitting sequence* for a theory $\mathcal{T} \in \mathfrak{T}$ is a monotone, continuous sequence $\langle U_\alpha \rangle_{\alpha < \mu}$ of splitting sets for $\mathcal{T}$ such that $\bigcup_{\alpha < \mu} U_\alpha = \mathfrak{U}$.

The definition of splitting sequences allows transfinite sequences in order to allow generalizations to programs with, e.g., infinite stable models [14] as shown below. As we will see, the definition of update operators for MKNF knowledge bases will not require such transfinite sequences.

Example 48. The following program illustrates the use of transfinite splitting sequences. Note that this program admits a splitting sequence in which U_α contains $\mathsf{exists\_ill\_defined\_number}$ only for a transfinite value of α.

$$\begin{aligned} \mathsf{exists\_ill\_defined\_number} &\leftarrow \mathsf{is\_number(X), even(X), odd(X)}. & \qquad \mathsf{is\_number(0)}. \\ \mathsf{is\_number(s(X))} &\leftarrow \mathsf{is\_number(X)}. & \mathsf{even(0)}. \\ \mathsf{even(s(s(X)))} &\leftarrow \mathsf{even(X)}. & \mathsf{odd(s(0))}. \\ \mathsf{odd(X)} &\leftarrow \sim\mathsf{even(X)}. & \end{aligned}$$

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 sequence 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 (given its continuity) that the binary operator $\uplus$ can be generalised to arbitrary subsets of $\mathbb{S}$, $\mathcal{X}_{\alpha+1}$ 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 49 *(Solution w.r.t. a splitting sequence).* Let S be a semantics for $(\mathfrak{T}, \mathbb{S})$ and $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha < \mu}$ a splitting sequence for a theory $\mathcal{T} \in \mathfrak{T}$. 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. $\mathcal{X}_0 \in [\![b_{U_0}(\mathcal{T})]\!]_{\mathsf{S}}$.
2. For any sequence ordinal α such that $\alpha + 1 < \mu$,
$$\mathcal{X}_{\alpha+1} \in \left[\!\!\left[e_{U_\alpha} \left(b_{U_{\alpha+1}}(\mathcal{T}), \biguplus_{\beta \leq \alpha} \mathcal{X}_\beta \right) \right]\!\!\right]_{\mathsf{S}} .$$
3. For any limit ordinal $\alpha < \mu$, $\mathcal{X}_\alpha = \mathbf{0}$.
4. $\biguplus_{\alpha < \mu} \mathcal{X}_\alpha \in \mathbb{S}$.

The Abstract Splitting Sequence Property is now a straightforward adaptation of the Abstract Splitting Set Property.

Definition 50 *(Abstract splitting sequence property).* We say that a semantics S for $(\mathfrak{T}, \mathbb{S})$ *satisfies the Abstract 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 $\mathcal{T}$,

$$[\![\mathcal{T}]\!]_{\mathsf{S}} = \left\{ \biguplus_{\alpha < \mu} \mathcal{X}_\alpha \;\middle|\; \langle \mathcal{X}_\alpha \rangle_{\alpha < \mu} \text{ is an } \mathsf{S}\text{-solution to } \mathcal{T} \text{ w.r.t. } \boldsymbol{U} \right\} .$$

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 [71], 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 considerably simplifies the splitting of ontologies, which usually contain axioms with an implicit universal quantifier.

4.2.1. Splitting properties for MKNF knowledge bases

We instantiate the Abstract Splitting Properties for MKNF knowledge bases as follows:

- The set of theories $\mathfrak{T}$ is the set of all MKNF knowledge bases over a language $\mathcal{L}$.
- The set of semantic structures $\mathbb{S}$ is the set of all MKNF interpretations.
- $\uplus$ is the set intersection $\cap$ with the neutral element $\mathbf{0} = \mathfrak{I}_{\mathcal{L}}$.
- The semantic function $[\![\cdot]\!]_{\mathsf{S}}$ returns all MKNF models of the argument MKNF knowledge base.

In this section, we restrict SNA interpretations (Definition 3) to those that interpret $\approx$ as the identity relation. A splitting set for an MKNF knowledge base is defined analogously 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.

In the following, given any MKNF formula, MKNF theory, ontology axiom, ontology, (generalised) literal, (MKNF) rule, (MKNF) program or MKNF knowledge base Δ, we denote by $\mathsf{pr}(\Delta)$ the set of predicate symbols occurring in Δ.

Definition 51 *(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 $\mathsf{pr}(\mathsf{H}_\pi) \cap U \neq \emptyset$, 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 formed using only predicate symbols from U. The top, on the other hand, contains the remaining ontology axioms and rules. Formally:

Definition 52 *(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

$$b_U(\mathcal{O}) = \{ \phi \in \mathcal{O} \mid \mathrm{pr}(\phi) \subseteq U \} \qquad \text{and} \qquad b_U(\mathcal{P}) = \{ \pi \in \mathcal{P} \mid \mathrm{pr}(\pi) \subseteq U \} \ .$$

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 $\mathrm{pr}(L) \subseteq U$ that is not satisfied in $\mathcal{X}$, and eliminate the remaining literals L with $\mathrm{pr}(L) \subseteq U$. This is formally captured as follows:

Definition 53 *(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

$$e_U(\mathcal{P}, \mathcal{X}) = \left\{ \mathsf{H}_\pi \leftarrow \{ L \in \mathsf{B}_\pi \mid \mathrm{pr}(L) \subseteq \mathfrak{P} \setminus U \}. \mid \pi \in t_U(\mathcal{P}) \wedge \mathcal{X} \models \kappa(\left\{ L' \in \mathsf{B}_\pi \mid \mathrm{pr}(L') \subseteq U \right\}) \right\}$$

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 semantics of MKNF knowledge bases satisfies both splitting properties:

Theorem 54 *(Splitting theorem for MKNF knowledge bases). The semantics of MKNF knowledge bases satisfies the Abstract Splitting Set Property and the Abstract Splitting Sequence Property.*

Proof. See Appendix C, page 84. □

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 2.

Example 55 *(Splitting the cargo imports knowledge base).* Consider the MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ presented in Fig. 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

$$\begin{aligned}
U_0 = \quad & \{ \ \mathsf{Commodity}/1, \mathsf{EdibleVegetable}/1, \mathsf{Tomato}/1, \mathsf{CherryTomato}/1, \\
& \quad \mathsf{GrapeTomato}/1, \mathsf{HTSCode}/2, \mathsf{HTSChapter}/2, \mathsf{HTSHeading}/2, \mathsf{Bulk}/1, \\
& \quad \mathsf{Prepackaged}/1, \mathsf{TariffCharge}/2, \mathsf{ShpmtCommod}/2, \mathsf{ShpmtImporter}/2, \\
& \quad \mathsf{ShpmtDeclHTSCode}/2, \mathsf{ShpmtProducer}/2, \mathsf{ShpmtCountry}/2 \ \} \ , \\
U_1 = U_0 \cup & \{ \ \mathsf{AdmissibleImporter}/1, \mathsf{SuspectedBadGuy}/1, \mathsf{ApprovedImporterOf}/2 \ \} \ , \\
U_2 = U_1 \cup & \{ \ \mathsf{RegisteredProducer}/2, \mathsf{EUCountry}/1, \mathsf{EURegisteredProducer}/1, \\
& \quad \mathsf{CommodCountry}/2, \mathsf{ExpeditableImporter}/2, \mathsf{LowRiskEUCommodity}/1 \ \} \ , \\
U_3 = U_2 \cup & \{ \ \mathsf{CompliantShpmt}/1, \mathsf{Random}/1, \mathsf{RandomInspection}/1, \mathsf{PartialInspection}/1, \\
& \quad \mathsf{FullInspection}/1 \ \} \ .
\end{aligned}$$

This splitting sequence divides $\mathcal{K}$ into the four layers shown in Fig. 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 both 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.

$* \; * \; * \;\; b_{U_0}(\mathcal{K}) \;\; * \; * \; *$

$\mathsf{Commodity} \equiv (\exists \mathsf{HTSCode}.\top)$
$\mathsf{CherryTomato} \equiv (\exists \mathsf{HTSCode}.\{ \text{‘07020020’} \})$
$\mathsf{GrapeTomato} \equiv (\exists \mathsf{HTSCode}.\{ \text{‘07020010’} \})$
$\mathsf{CherryTomato} \sqsubseteq \mathsf{Tomato}$
$\mathsf{CherryTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{ \$0 \})$
$\mathsf{GrapeTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{ \$40 \})$
$\mathsf{CherryTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{ \$50 \})$
$\mathsf{GrapeTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{ \$100 \})$
$\mathsf{EdibleVegetable} \equiv (\exists \mathsf{HTSChapter}.\{ \text{‘07’} \})$
$\mathsf{Tomato} \equiv (\exists \mathsf{HTSHeading}.\{ \text{‘0702’} \})$
$\mathsf{Tomato} \sqsubseteq \mathsf{EdibleVegetable}$
$\mathsf{GrapeTomato} \sqsubseteq \mathsf{Tomato}$
$\mathsf{CherryTomato} \sqcap \mathsf{GrapeTomato} \sqsubseteq \bot$
$\mathsf{Bulk} \sqcap \mathsf{Prepackaged} \sqsubseteq \bot$

$\mathsf{ShpmtCommod}(s_1, c_1)$
$\mathsf{ShpmtImporter}(s_1, i_1)$
$\mathsf{ShpmtCommod}(s_2, c_2)$
$\mathsf{ShpmtImporter}(s_2, i_2)$
$\mathsf{ShpmtCountry}(s_2, \mathit{portugal})$
$\mathsf{ShpmtCommod}(s_3, c_3)$
$\mathsf{ShpmtImporter}(s_3, i_3)$
$\mathsf{ShpmtCountry}(s_3, \mathit{portugal})$
$\mathsf{ShpmtDeclHTSCode}(s_1, \text{‘07020020’})$
$\mathsf{CherryTomato}(c_1) \quad \mathsf{Bulk}(c_1)$
$\mathsf{ShpmtDeclHTSCode}(s_2, \text{‘07020020’})$
$\mathsf{CherryTomato}(c_2) \quad \mathsf{Prepackaged}(c_2)$
$\mathsf{ShpmtDeclHTSCode}(s_3, \text{‘07020010’})$
$\mathsf{GrapeTomato}(c_3) \quad \mathsf{Bulk}(c_3)$
$\mathsf{ShpmtProducer}(s_3, p_1)$

$* \; * \; * \;\; t_{U_0}(b_{U_1}(\mathcal{K})) \;\; * \; * \; *$

$\mathsf{AdmissibleImporter}(\mathbf{x}) \leftarrow {\sim}\mathsf{SuspectedBadGuy}(\mathbf{x}).$
$\mathsf{ApprovedImporterOf}(i_2, \mathbf{x}) \leftarrow \mathsf{EdibleVegetable}(\mathbf{x}).$
$\mathsf{ApprovedImporterOf}(i_3, \mathbf{x}) \leftarrow \mathsf{GrapeTomato}(\mathbf{x}).$
$\mathsf{SuspectedBadGuy}(i_1).$

$* \; * \; * \;\; t_{U_1}(b_{U_2}(\mathcal{K})) \;\; * \; * \; *$

$\mathsf{EURegisteredProducer} \equiv (\exists \mathsf{RegisteredProducer}.\mathsf{EUCountry})$
$\mathsf{LowRiskEUCommodity} \equiv (\exists \mathsf{ExpeditableImporter}.\top) \sqcap (\exists \mathsf{CommodCountry}.\mathsf{EUCountry})$
$\mathsf{CommodCountry}(\mathbf{x}, \mathbf{y}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{z}, \mathbf{x}), \mathsf{ShpmtCountry}(\mathbf{z}, \mathbf{y}).$
$\mathsf{ExpeditableImporter}(\mathbf{x}, \mathbf{y}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{z}, \mathbf{x}), \mathsf{ShpmtImporter}(\mathbf{z}, \mathbf{y}),$
$\qquad \mathsf{AdmissibleImporter}(\mathbf{y}), \mathsf{ApprovedImporterOf}(\mathbf{y}, \mathbf{x}).$

$\mathsf{RegisteredProducer}(p_1, \mathit{portugal})$
$\mathsf{RegisteredProducer}(p_2, \mathit{slovakia})$
$\mathsf{EUCountry}(\mathit{portugal})$
$\mathsf{EUCountry}(\mathit{slovakia})$

$* \; * \; * \;\; t_{U_2}(b_{U_3}(\mathcal{K})) \;\; * \; * \; *$

$\mathsf{CompliantShpmt}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{HTSCode}(\mathbf{y}, \mathbf{z}), \mathsf{ShpmtDeclHTSCode}(\mathbf{x}, \mathbf{z}).$
$\mathsf{RandomInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{Random}(\mathbf{y}).$
$\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{RandomInspection}(\mathbf{x}).$
$\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), {\sim}\mathsf{LowRiskEUCommodity}(\mathbf{y}).$
$\mathsf{FullInspection}(\mathbf{x}) \leftarrow {\sim}\mathsf{CompliantShpmt}(\mathbf{x}).$
$\mathsf{FullInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtCommod}(\mathbf{x}, \mathbf{y}), \mathsf{Tomato}(\mathbf{y}), \mathsf{ShpmtCountry}(\mathbf{x}, \mathit{slovakia}).$

Fig. 2. Splitting Sequence Layers of the MKNF knowledge base for Cargo Imports.

4.2.2. Ontology updates

As shown, MKNF Knowledge Bases have the splitting properties: in this section we show that in the case of an ontology Winslett's update operator also has the Abstract Splitting Properties. Instantiation of the generic splitting properties for a given first-order update operator $\diamond$ is done as follows:

- The set of theories $\mathfrak{T}$ contains finite sequences of first-order theories.
- The set of semantic structures $\mathcal{S} = 2^{\mathcal{I}_{\mathcal{L}}}$ contains all sets of first-order interpretations.
- As before with MKNF knowledge bases, $\uplus$ is the set intersection $\cap$ with the neutral element $\mathbf{0} = \mathcal{I}_{\mathcal{L}}$.
- The semantic function $[\![\cdot]\!]_{\mathsf{S}}$ is defined by $[\![\boldsymbol{T}]\!]_{\mathsf{S}} = \left\{ \, [\![\diamond \boldsymbol{T}]\!] \, \right\}$.

The splitting set, top, bottom and reduct of a first-order theory are defined analogously 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 56 *(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 $\mathrm{pr}(\phi) \cap U \neq \emptyset$, then $\mathrm{pr}(\phi) \subseteq U$;
- *splitting set for $\boldsymbol{T}$* if for every $i < n$, U is a splitting set for $\mathcal{T}_i$.

The *bottom* and *top of $\mathcal{T}$ relative to U* are the theories

$$b_U(\mathcal{T}) = \{ \phi \in \mathcal{T} \mid \mathrm{pr}(\phi) \subseteq U \} \qquad \text{and} \qquad t_U(\mathcal{T}) = \mathcal{T} \setminus b_U(\mathcal{T}) \ .$$

The *bottom* and *top of $\boldsymbol{T}$ relative to U* are the sequences of theories

$$b_U(\boldsymbol{T}) = \langle b_U(\mathcal{T}_i) \rangle_{i<n} \qquad \text{and} \qquad t_U(\boldsymbol{T}) = \langle t_U(\mathcal{T}_i) \rangle_{i<n} \ .$$

Given some $\mathcal{X} \in \mathfrak{M}$, 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 directly use the Abstract Splitting Properties from Section 4.1. Furthermore, since every sequence of first-order theories has a single set of models, the resulting properties are simpler than their general form (which allows a set of sets of models). Particularly, given a sequence of first-order theories $\boldsymbol{T}$ and a splitting set U for $\boldsymbol{T}$, the Abstract Splitting Set Property requires that

$$[\![\diamond \boldsymbol{T}]\!] = [\![\diamond b_U(\boldsymbol{T})]\!] \cap [\![\diamond t_U(\boldsymbol{T})]\!] \ .$$

Similarly, given a splitting sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$ for $\boldsymbol{T}$, the Abstract Splitting Sequence Property requires that

$$[\![\diamond \boldsymbol{T}]\!] = [\![\diamond b_{U_0}(\boldsymbol{T})]\!] \cap \bigcap_{\alpha+1<\mu} [\![\diamond t_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{T}))]\!] \ .$$

If we pick Winslett's first-order operator as $\diamond$, then both of these properties are satisfied.

Theorem 57 *(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 Abstract Splitting Set Property and Abstract Splitting Sequence Property.*

Proof. See Appendix C, page 87. □

4.2.3. Rule updates

To complete the discussion of splitting properties we turn to the special case of updating program rules and show that the RD semantics (Section 2.6) observes the splitting properties. Given a rule update semantics S, the splitting properties for DLPs can be derived from the generic ones as follows:

- The set of theories $\mathfrak{T}$ contains all DLPs.
- The set of semantic structures $\mathfrak{S}$ coincides with the set of all LP 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-models (i.e., stable models) of the argument DLP.

Since rule update semantics work with LP interpretations instead of MKNF interpretations, we define the program reduct w.r.t. an LP interpretation as the program reduct w.r.t. the corresponding MKNF interpretation. Note that despite the different definition, the resulting concept is in line with the same notion in [71].

Definition 58 *(Reduct of a program w.r.t. a splitting set).* Let $\mathcal{P}$ be a program, $U \subseteq \mathfrak{P}$ a set of predicate symbols, J an LP 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 59 *(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 LP 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 Abstract Splitting Properties of 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. The RD-semantics for rule updates (Section 2.6), as well as many other rule update semantics that generalise the stable model semantics, e.g. [6,37,66,78], naturally satisfy both splitting properties:

Theorem 60 *(Splitting theorem for rule updates). The* RD*-semantics for rule updates satisfies the Abstract Splitting Set Property and the Abstract Splitting Sequence Property.*

Proof (sketch). See Appendix C, page 87. □

5. Layered dynamic MKNF knowledge bases

We are now ready to utilise the Abstract Splitting Properties as a foundation for a hybrid update semantics. More particularly, our semantics modularly combines a given first-order update operator $\diamond$ together with a given rule update semantics S and uses them to update DMKBs consisting of one or more ontology and rule layers, each feeding information into subsequent layers. Updates of each layer, depending on whether it is an ontology or a rule layer, are handled by $\diamond$ and S, respectively, and the resulting models are then used to assemble an overall dynamic MKNF model.

In the rest of this section we implicitly work under the following assumptions.

- As in Section 3, we assume that all MKNF rules are ground: in our case, with respect to $\mathfrak{C}$, the set of constants in $\mathcal{L}$.
- Unlike Section 3, we now allow rules with default negation in their heads. However, we do not allow rules to contain generalised atoms, as, to our knowledge, rules with generalised atoms are not yet supported by any rule update semantics.
- As in Section 4.2, we restrict SNA interpretations (Definition 3) to those that interpret $\approx$ as the identity relation.

With these assumptions in mind, we first define the concepts underlying splitting properties to identify a class of layered DMKBs to which we assign an update semantics (Section 5.1). Nevertheless, this generic semantics depends on a particular splitting sequence and in order for it to be splitting-independent, three properties of $\diamond$ and S need to be assumed: the Abstract Splitting Sequence Property and two basic properties of update operations. After defining them formally, we prove splitting-independence of the hybrid update semantics (Section 5.2). Then we turn to its other formal properties and illustrate how it can be applied to the scenario described in Example 2 (Section 5.3). The relevant proofs can be found in Appendix D.[20]

5.1. Splitting-based updates of MKNF knowledge bases

In the following we identify a class of DMKBs for which splitting enables us to perform updates by using a given first-order update operator together with a given rule update semantics.

Throughout the remainder of this section 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), or
b) arbitrary rules but no ontological axioms whatsoever.

Formally:

Definition 61 *(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 62 *(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 $J \in [\![\langle \mathcal{P}_i \rangle_{i<n}]\!]_{\mathsf{S}}$.

[20] A preliminary version of this work has been published in [91]. The present contribution is more general since it allows for various combinations of first-order and rule update semantics, not just for one fixed pair as in [91].

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. 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 63 *(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 $\mathcal{X} \in \mathcal{M}$, 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 64 *(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 $\mathcal{X} \in \mathcal{M}$, $e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \mathcal{X})$ 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. However, for the sake of completeness, the definition of a $(\diamond, \mathsf{S})$*-dynamic MKNF model of $\boldsymbol{K}$ w.r.t. a layering splitting sequence* is provided in Appendix D, Definition 143.

5.2. Independence of splitting sequence

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 splitting-independent. 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:* Both $\diamond$ and S must conserve the language, i.e. the models of syntactically independent knowledge base sequences must be independent. If this were not the case, then $\diamond$ and S might interfere with one another when used to update syntactically unrelated layers of the DMKB. This property is formalised below.
3. *Fact update:* Finally, since DMKBs consisting of a sequence of consistent sets of facts are classified by Definition 61 as both ontology- and rule-based, their semantics is given both by $\diamond$ and by S. This ambivalence is unavoidable if we want to allow both an ontology and a rule layer to contain only facts, or simply to be empty. Nevertheless, if the semantics assigned to such sequences of fact bases by $\diamond$ differs from that assigned by S, the resulting hybrid update semantics cannot generalise either $\diamond$ or S. In order to avoid such anomalies, we assume that both $\diamond$ and S respect fact update, as defined below.

Both language conservation and fact update are basic properties of update operations. For a rule update semantics S they can be formulated straightforwardly as follows:

Definition 65 *(Language conservation and fact update 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 $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ and every S-model J of $\boldsymbol{P}$, if $\mathrm{pr}(\mathcal{P}_i) \subseteq A$ for all $i < n$, then $\mathrm{pr}(J) \subseteq A$;
- *respects fact update* if for every finite sequence of consistent sets of facts $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, the unique S-model of $\boldsymbol{P}$ is the LP interpretation

$$\left\{ l \in \mathit{Lits}_{\mathsf{G}} \;\middle|\; \exists j < n : (l.) \in \mathcal{P}_j \wedge (\forall i : j < i < n \Longrightarrow \left\{ \bar{l}., {\sim}l. \right\} \cap \mathcal{P}_i = \emptyset) \right\} .^{21}$$

[21] Recall from Section 2.2 that $\mathit{Lits}_{\mathsf{G}}$ is the set of ground objective literals formed over $\mathcal{L}$; $\bar{l}$ is the complement of the objective literal l, and ${\sim}l$ is its default negation.

Intuitively, language conservation for rule updates requires that all atoms true in an S-stable model of a DLP $\boldsymbol{P}$ must appear in at least one of its rules, which is a necessary condition for the atom to be properly justified. On the other hand, fact update enforces a principle of inertia that forms the basis for belief update operators such as Winslett's. Both of these properties are satisfied by a wide range of rule update semantics, including the RD-semantics.

Theorem 66 *(Language conservation and fact update for* RD*-semantics). The* RD*-semantics for rule updates conserves the language and respects fact update.*

Proof (sketch). See Appendix D, page 89. □

The formalisation of language conservation for first-order updates relies on the notion of *interpretation restriction* and the related concept of *saturated set of interpretations*. Intuitively, given a set of predicate symbols A, a set of interpretations $\mathcal{M} \in \mathfrak{M}$ is *saturated relative to* A if it only contains knowledge about predicate symbols from A.

Definition 67 *(Interpretation restriction and saturated interpretation).* Let A be a set of predicate symbols, $I \in \mathfrak{I}_{\mathcal{L}}$ and $\mathcal{M} \in \mathfrak{M}$. The *restriction of* I *to* A is an interpretation $I^{[A]}$ such that for every ground atom p,

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

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

Furthermore, we say that $\mathcal{M}$ is *saturated relative to* A if for every interpretation $I \in \mathfrak{I}_{\mathcal{L}}$,

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

Example 68. Suppose a language $\mathcal{L}$ contains only the predicate symbols $p/0$, $q/0$, and $r/0$. Let $A = \{ p/0, q/0 \}$ and let $\mathcal{M} = \{ \{ p, q \}, \{ p, q, r \} \}$. Then $\mathcal{M}^{[A]} = \{ \{ p, q \} \}$ and it is not difficult to verify that $\mathcal{M}$ is saturated relative to A. However, $\mathcal{M}$ is not saturated with respect to $B = \{ p/0 \}$. Since $\mathcal{M}^{[B]} = \{ \{ p \} \}$, we see that even though $I = I^{[B]} = \{ p \} \in \mathcal{M}^{[B]}$, it is not the case that $I \in \mathcal{M}$.

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

Proof. See Appendix D, page 89. □

In other words, the set of models of a single first-order theory conserves its language. Language conservation for an update operator $\diamond$ extends the same property to sequences of first-order theories. Also, for first-order theories, fact update is formulated in essentially the same way as for rule updates.

Definition 70 *(Language conservation and fact update for ontology updates).* 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 $i < n$, $\mathrm{pr}(\mathcal{T}_i) \subseteq A$, $[\![\diamond \boldsymbol{T}]\!]$ is saturated relative to A;
- *respects fact update* if for every finite sequence of consistent sets of ground objective literals $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$,

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

Since Winslett's operator treats all predicate symbols in a uniform manner, it naturally satisfies both these properties.

Theorem 71 *(Language conservation and fact update for Winslett's operator). Winslett's first-order update operator $\diamond_{\mathsf{W}}$ conserves the language and respects fact update.*

Proof. See Appendix D, page 90. □

The previous results show that if we consider Winslett's operator for performing ontology updates and the RD-semantics for performing rule updates, then they satisfy the splitting properties (cf. Theorems 57 and 60), conserve the language and respect fact update. Furthermore, these three basic assumptions about the first-order update operator $\diamond$ and rule update semantics S are sufficient to guarantee that $(\diamond, \mathsf{S})$-dynamic MKNF models are independent of the choice of the layering splitting sequence.

Proposition 72 *(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 D, page 93. □

We can now safely introduce the dynamic MKNF model of a layered DMKB, independent of a particular layering splitting sequence:

Definition 73 *(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 $\langle \mathcal{P} \rangle$ (i.e., the sequence containing the single element, $\mathcal{P}$) is a layering splitting sequence for any basic DMKB, it follows from Proposition 72 that the above definition is compatible with the previously defined semantics for basic DMKBs (cf. Definition 62).

Finally, we define a consequence relation by considering the credulous consequence of $(\diamond, \mathsf{S})$-dynamic MKNF models.

Definition 74 *(Consequence relation).* Suppose that both $\diamond$ and S have the splitting sequence property, conserve the language and respect fact update, let $\boldsymbol{K}$ be a layered DMKB and $\mathcal{T}$ an MKNF theory. We say that $\boldsymbol{K}$ $(\diamond, \mathsf{S})$-*MKNF entails* $\mathcal{T}$, denoted by $\boldsymbol{K} \models^{\diamond,\mathsf{S}}_{\mathsf{MKNF}} \mathcal{T}$, if $\mathcal{M} \models \mathcal{T}$ for some $(\diamond, \mathsf{S})$-dynamic MKNF model $\mathcal{M}$ of $\boldsymbol{K}$.

5.3. Properties and use

The purpose of this subsection is twofold. First, we establish other formal properties of the hybrid update semantics that has just been defined, 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 fully in line with the hybrid update semantics defined in Section 3.

Our second goal is to illustrate its usefulness by considering updates of the MKNF knowledge base about Cargo Imports presented in Example 2.

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.

5.3.1. Fidelity of the hybrid update semantics

First we 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, including [6,7,13,31,37,65,66,78,80,98].

Definition 75 *(Faithfulness to stable models and primacy of new information).* Let S be a rule update semantics. We say that S

- *is faithful to the stable model semantics* if for every program $\mathcal{P}$, an LP interpretation J is an S-model of the DLP $\langle \mathcal{P} \rangle$ if and only if J is a stable model of $\mathcal{P}$;
- *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 model semantics.

Theorem 76 *(Faithfulness w.r.t. MKNF knowledge bases). Suppose that S is faithful to the stable model 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 D, page 96. □

An immediate consequence of this result and of the basic properties of MKNF knowledge bases (Proposition 15) 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 77 *(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 D, page 96. □

Theorem 78 *(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 LP interpretation corresponding to* $\mathcal{M}$ *is an* S*-model of* $\langle \mathcal{P}_i \rangle_{i<n}$.

Proof. See Appendix D, page 96. □

As with the update semantics in Section 3, assuming that $\diamond$ satisfies (FO1) and S respects primacy of new information, the resulting hybrid semantics also respects it.

Theorem 79 *(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 D, page 96. □

Finally, the update semantics is compatible with the semantics from Section 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 80 *(Compatibility with update semantics from Section 3). Suppose that* $\diamond$ *satisfies (FO2.*$\top$*) and (FO8.2) and that* S *is faithful to the stable model 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 D, page 101. □

5.3.2. An extended example

Finally, 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 81 *(Updating the cargo imports knowledge base).* The MKNF knowledge base $\mathcal{K}$ in Fig. 1 has a single MKNF model $\mathcal{M}$. Due to the splitting theorem for MKNF knowledge bases (cf. Theorem 54), $\mathcal{M}$ coincides with $\mathcal{X}_0 \cap \mathcal{X}_1 \cap \mathcal{X}_2 \cap \mathcal{X}_3$ where $\mathcal{X}_i$ is the MKNF model of the i-th layer of $\mathcal{K}$, as presented in Fig. 2. We shortly summarise what is entailed by each of these MKNF models.

The first layer of $\mathcal{K}$, $b_{U_0}(\mathcal{K})$, contains a number of ABox assertions about shipments s_1, s_2, s_3 which differ in the kind of tomatoes or in their packaging. All of these assertions are entailed by $\mathcal{X}_0$ and, besides that, using the TBox assertions we can derive that the cargo within each shipment (c_1, c_2 and c_3, respectively) belongs to the concept Tomato as well as EdibleVegetable, and that each cargo is assigned a different tariff charge:

$$\begin{aligned} \mathcal{X}_0 \models \{ \; & \mathsf{CherryTomato}(c_1), \; \mathsf{Tomato}(c_1), \; \mathsf{EdibleVegetable}(c_1), \; \mathsf{TariffCharge}(c_1, \$0), \\ & \mathsf{CherryTomato}(c_2), \; \mathsf{Tomato}(c_2), \; \mathsf{EdibleVegetable}(c_2), \; \mathsf{TariffCharge}(c_2, \$50), \\ & \mathsf{GrapeTomato}(c_3), \; \mathsf{Tomato}(c_3), \; \mathsf{EdibleVegetable}(c_3), \; \mathsf{TariffCharge}(c_3, \$40) \} \; . \end{aligned}$$

Using this information, the rules in the second layer of $\mathcal{K}$, $t_{U_0}(b_{U_1}(\mathcal{K}))$, derive information about admissible and approved importers, namely:

$$\begin{aligned} \mathcal{X}_1 \models \{ \; & \mathsf{SuspectedBadGuy}(i_1), \; \mathsf{ApprovedImporterOf}(i_2, c_1), \\ & \mathsf{AdmissibleImporter}(i_2), \; \mathsf{ApprovedImporterOf}(i_2, c_2), \\ & \mathsf{AdmissibleImporter}(i_3), \; \mathsf{ApprovedImporterOf}(i_2, c_3), \\ & \qquad \mathsf{ApprovedImporterOf}(i_3, c_3) \} \; . \end{aligned}$$

Turning to the third layer, the rules and TBox axioms of $t_{U_1}(b_{U_2}(\mathcal{K}))$, entail some simple geographic information along with the following:

$$\begin{aligned} \mathcal{X}_2 \models \{ \; & \mathsf{ExpeditableImporter}(c_2, i_2), \; \mathsf{LowRiskEUCommodity}(c_2), \\ & \mathsf{ExpeditableImporter}(c_3, i_3), \; \mathsf{LowRiskEUCommodity}(c_3), \\ & \mathsf{EURegisteredProducer}(p_1) \} \; . \end{aligned}$$

$$* \; * \; * \quad b_{U_0}(\mathcal{K}') \quad * \; * \; *$$

$\mathsf{Commodity} \equiv (\exists \mathsf{HTSCode}.\top)$
$\mathsf{CherryTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020020'}\})$
$\mathsf{GrapeTomato} \equiv (\exists \mathsf{HTSCode}.\{\text{'07020010'}\})$
$\mathsf{CherryTomato} \sqsubseteq \mathsf{Tomato}$
$\mathsf{CherryTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$0\})$
$\mathsf{GrapeTomato} \sqcap \mathsf{Bulk} \equiv (\exists \mathsf{TariffCharge}.\{\$40\})$
$\mathsf{CherryTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$50\})$
$\mathsf{GrapeTomato} \sqcap \mathsf{Prepackaged} \equiv (\exists \mathsf{TariffCharge}.\{\$100\})$
$\mathsf{EdibleVegetable} \equiv (\exists \mathsf{HTSChapter}.\{\text{'07'}\})$
$\mathsf{Tomato} \equiv (\exists \mathsf{HTSHeading}.\{\text{'0702'}\})$
$\mathsf{Tomato} \sqsubseteq \mathsf{EdibleVegetable}$
$\mathsf{GrapeTomato} \sqsubseteq \mathsf{Tomato}$
$\mathsf{CherryTomato} \sqcap \mathsf{GrapeTomato} \sqsubseteq \bot$
$\mathsf{Bulk} \sqcap \mathsf{Prepackaged} \sqsubseteq \bot$

$\mathsf{GrapeTomato}(c_1)$

$$* \; * \; * \quad t_{U_0}(b_{U_1}(\mathcal{K}')) \quad * \; * \; *$$

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

$$* \; * \; * \quad t_{U_1}(b_{U_2}(\mathcal{K}')) \quad * \; * \; *$$

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

$\neg\mathsf{LowRiskEUCommodity}(c_3)$

$$* \; * \; * \quad t_{U_2}(b_{U_3}(\mathcal{K}')) \quad * \; * \; *$$

$\sim\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtProducer}(\mathbf{x}, \mathbf{y}), \mathsf{EURegisteredProducer}(\mathbf{y}).$

Fig. 3. Layers of the update to the MKNF knowledge base for Cargo Imports.

The HTS codes of commodities inside all three shipments match the declared HTS codes, so all shipments are compliant according to the rules in the last layer, $t_{U_2}(b_{U_3}(\mathcal{K}))$. Besides, there is no expeditable importer for c_1, so it is not considered a low risk commodity and the rules in the last layer imply that a partial inspection of s_1 will take place:

$$\mathcal{X}_3 \models \{\ \mathsf{CompliantShipment}(s_1),\ \mathsf{CompliantShipment}(s_2),\ \mathsf{CompliantShipment}(s_3),\ \mathsf{PartialInspection}(s_1)\ \}\ .$$

Now we 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 $\mathcal{O}$, in order to keep them static; and $\mathcal{P}'$ contains the following rules[22]:

$$\sim\mathsf{ApprovedImporterOf}(i_2, \mathbf{x}) \leftarrow \mathsf{Tomato}(\mathbf{x}).$$
$$\sim\mathsf{PartialInspection}(\mathbf{x}) \leftarrow \mathsf{ShpmtProducer}(\mathbf{x}, \mathbf{y}), \mathsf{EURegisteredProducer}(\mathbf{y}).$$

Note that the splitting sequence $\boldsymbol{U}$ defined in Example 55 is a layering splitting sequence for the DMKB $\boldsymbol{K} = \langle \mathcal{K}, \mathcal{K}' \rangle$. The four layers of $\mathcal{K}$ and $\mathcal{K}'$, listed in Figs. 2 and 3, respectively, are used to form the corresponding layers of $\boldsymbol{K}$. The first layer, $b_{U_0}(\boldsymbol{K}) = \langle b_{U_0}(\mathcal{K}), b_{U_0}(\mathcal{K}') \rangle$, still contains only ontology axioms and so is ontology-based. The second and fourth layers, $t_{U_0}(b_{U_1}(\boldsymbol{K})) = \langle t_{U_0}(b_{U_1}(\mathcal{K})), t_{U_0}(b_{U_1}(\mathcal{K}')) \rangle$ and $t_{U_2}(b_{U_3}(\boldsymbol{K})) = \langle t_{U_2}(b_{U_3}(\mathcal{K})), t_{U_2}(b_{U_3}(\mathcal{K}')) \rangle$, contain only rules and so are rule-based. Finally, the third layer, $t_{U_1}(b_{U_2}(\boldsymbol{K})) = \langle t_{U_1}(b_{U_2}(\mathcal{K})), t_{U_1}(b_{U_2}(\mathcal{K}')) \rangle$, contains a mixture of rules and ontology

[22] We assume that all rule variables are grounded prior to applying our theory.

axioms but all the rules have positive heads and predicate symbols of all body literals belong to U_1, so the reduct of the third layer will necessarily be ontology-based. This implies that $\boldsymbol{K}$ is a layered DMKB.

In order to arrive at a $(\diamond_W, \mathsf{RD})$-dynamic MKNF model of $\boldsymbol{K}$ with respect to $\boldsymbol{U}$, a $(\diamond_W, \mathsf{RD})$-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 $b_{U_0}(\boldsymbol{K})$. Due to the TBox axioms, the cargo c_1 is no longer classified as cherry tomatoes, and its HTS code as well as the tariff charge change. Note that the conflict between old and new knowledge is properly resolved by Winslett's operator:

$$\begin{aligned}\mathcal{X}'_0 \models \{ \; & \neg\mathsf{CherryTomato}(c_1), && \mathsf{GrapeTomato}(c_1), \\ & \neg\mathsf{HTSCode}(c_1, \text{'07020020'}), && \mathsf{HTSCode}(c_1, \text{'07020010'}), \\ & \neg\mathsf{TariffCharge}(c_1, \$0), && \mathsf{TariffCharge}(c_1, \$40) \, \} \; .\end{aligned}$$

Subsequently, the RD-semantics is used to find the model $\mathcal{X}'_1$ of the reduct of the second layer w.r.t. $\mathcal{X}'_0$, i.e. $e_{U_0}(b_{U_1}(\boldsymbol{K}), \mathcal{X}'_0)$. This results in i_2 no longer being an approved importer for any of the cargo in the knowledge base. As before, the conflict that arose is resolved by the RD-semantics:

$$\begin{aligned}\mathcal{X}'_1 \models \{ \; & \sim\mathsf{ApprovedImporterOf}(i_2, c_1), && \sim\mathsf{ApprovedImporterOf}(i_2, c_2), \\ & && \sim\mathsf{ApprovedImporterOf}(i_2, c_3) \, \} \; .\end{aligned}$$

Given $\mathcal{X}'_0$ and $\mathcal{X}'_1$, we form the reduct of the third layer: $e_{U_1}(b_{U_2}(\boldsymbol{K}), \mathcal{X}'_0 \cap \mathcal{X}'_1)$. It follows that i_2 is no longer an expeditable importer of c_2 and, as a consequence, c_2 is no longer considered a low risk commodity. Furthermore, due to the explicit update, c_3 is also no longer filed as a low risk commodity. The conflicting situation is again resolved by Winslett's operator and results in the $(\diamond_W, \mathsf{RD})$-dynamic MKNF model $\mathcal{X}'_2$ such that:

$$\begin{aligned}\mathcal{X}'_2 \models \{ \; & \sim\mathsf{ExpeditableImporter}(c_2, i_2), && \sim\mathsf{LowRiskEUCommodity}(c_2), \\ & \neg\mathsf{LowRiskEUCommodity}(c_3), && \mathsf{EURegisteredProducer}(p_1) \, \} \; .\end{aligned}$$

Finally, due to all the changes in the three previous layers, the rules in the reduct of the fourth layer, $e_{U_2}(b_{U_3}(\boldsymbol{K}), \mathcal{X}'_0 \cap \mathcal{X}'_1 \cap \mathcal{X}'_2)$, imply that shipment s_1 is no longer compliant and, as a consequence, should be fully inspected. Also, shipment s_2 needs to be partially expected because c_2 is now not a low risk commodity. Furthermore, even though c_3 is also not a low risk commodity, it is not going to be partially inspected because of the rule update in 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:

$$\begin{aligned}\mathcal{X}'_3 \models \{ \; & \sim\mathsf{CompliantShpmt}(s_1), && \mathsf{FullInspection}(s_1), \\ & \mathsf{PartialInspection}(s_2), && \sim\mathsf{PartialInspection}(s_3), \, \} \; .\end{aligned}$$

Note that the sequence of MKNF interpretations $\langle \mathcal{X}'_0, \mathcal{X}'_1, \mathcal{X}'_2, \mathcal{X}'_3 \rangle$ is a solution to $\boldsymbol{K}$ w.r.t. the splitting sequence $\boldsymbol{U}$ and the unique $(\diamond_W, \mathsf{RD})$-dynamic MKNF model assigned to $\boldsymbol{K}$ is the MKNF interpretation $\mathcal{M}' = \mathcal{X}'_0 \cap \mathcal{X}'_1 \cap \mathcal{X}'_2 \cap \mathcal{X}'_3$.

6. Complexity of update query answering

The preceding sections have presented two complementary approaches to updating MKNF knowledge bases: DMKBs with static rules in Section 3, and layered DMKBs in Section 5. In this section, after informally reviewing previous results on the complexity of entailment from MKNF knowledge bases, we present a basic complexity result for each approach. We assume the basic terminology of computational complexity as can be found in e.g., [79]. As notation, if Ω_1 and Ω_2 are complexity classes, then $\Omega_1^{\Omega_2}$ denotes the class of problems that can be solved by a Turing machine that is complete for class Ω_1 which makes use of an oracle that is complete for problems in class Ω_2.

6.1. Complexity of entailment from MKNF knowledge bases

The original paper on MKNF knowledge bases, [77] presented a variety of complexity results concerning entailment by a MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$. More specifically, they considered the problem of determining whether $\mathcal{K} \models \mathbf{K}\, A$, where A is a ground generalised atom in an atom base associated with $\mathcal{K}$, and given that the underlying description logic of $\mathcal{O}$ is *DL*.

The first measure of complexity considered, termed combined complexity, is based on the size of (non-ground) $\mathcal{K}$ plus the size of A. Since no assumptions are made about $\mathcal{P}$ – which may be non-ground – when *DL* is of low complexity, the cost of combined complexity is dominated by the cost of grounding $\mathcal{P}$ which is exponential in the size of $\mathcal{P}$. Even for more expressive cases of *DL* such as $\mathcal{SHOIN}$ or $\mathcal{SROIQ}$ which underlay specifications of OWL, the complexity of entailment may be increased due to the generality of MKNF knowledge bases together with the weakness of assumptions made by combined complexity.

A second measure is data complexity, which makes stronger assumptions, including various boundedness conditions about size of A, $\mathcal{P}$ and $\mathcal{O}$, and the size of rules in $\mathcal{P}$, and that the Abox of $\mathcal{O}$ consists only of atomic formulas. Under such conditions, the data complexity of entailment for many description logics is either in Ptime or is co-NP complete. When such description logics are used in an MKNF knowledge base $\mathcal{K}$ the complexity of entailment is co-NP complete, or Π_2^P complete, respectively. In the case of stratified programs, which have a single stable model, the complexity is Ptime-complete or is Δ_2^P-complete respectively.[23]

6.2. Complexity of entailment from dynamic MKNF knowledge bases

Since entailment by MKNF knowledge bases is not decidable in general, both the results mentioned above and those presented below apply only to *admissible* MKNF knowledge bases, a class that we now define. As discussed informally in Section 2.4, two conditions suffice for entailment by a (finite) MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ to be decidable. First, the description logic over which $\mathcal{O}$ is formulated must be decidable; and second, $\mathcal{P}$ must be DL-safe, which ensures that $\mathsf{ground}(\mathcal{P})$ is finite. In [77], these conditions are stated formally by the definition of an admissible MKNF knowledge base. Here, we extend this definition to ground DMKBs (sequences of ground MKNF knowledge bases, Definition 19):

Definition 82 *(Admissibility).* Let $\mathcal{DL}$ be a description logic, $\mathcal{B}$ a generalised atom base expressible in $\mathcal{DL}$, and $\mathcal{H} \subseteq \mathcal{B}$. $\mathcal{DL}$, $\mathcal{B}$, and $\mathcal{H}$ are *admissible* if, for each $\mathcal{O} \in \mathcal{DL}$, each finite set $S \subseteq \mathcal{H}$ of ground generalised atoms, each finite set N of assertions of the form $a \not\approx b$, and each generalised atom $\xi \in \mathcal{B}$, checking whether $\mathcal{O} \cup S \cup N \models \kappa(\xi)$ is decidable.

Let $\mathcal{DL}$, $\mathcal{B}$ and $\mathcal{H}$ be admissible. A program $\mathcal{P}$ over $\mathcal{B}$ is *admissible* if it is ground and finite and each rule in $\mathcal{P}$ contains a generalised literal over $\mathcal{H}$ in its head. A DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ is *admissible* if each $\mathcal{P}_i$ is admissible. An MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ over $\mathcal{DL}$ and $\mathcal{B}$ is admissible if it is ground, DL-safe and finite, and $\mathcal{P}$ is admissible. A DMKB $\langle \mathcal{K}_i \rangle_{i<n}$, is admissible if each $\mathcal{K}_i$ is admissible.[24]

In the remainder of this section, we assume that all ontologies and programs are over an admissible description logic $\mathcal{DL}$ and generalised atom base $\mathcal{B}$.

Next, we define the problem of query answering for updates to the ontology. The complexity of these problems will serve as parameters to the complexity of updating MKNF knowledge bases.

Definition 83 *(Query answering for ontology updates).* Let $\diamond$ be a first-order update operator and let Ω be a complexity class. We say that *query answering for* $\diamond$ *belongs to* Ω if for every sequence of ontologies $\langle \mathcal{O}_i \rangle_{i<n}$ and every finite set of ground generalised literals S, Ω contains the problem of deciding whether

$$[\![\mathcal{O}_0 \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \models \kappa(S) \ .$$

6.2.1. DMKBs with static rules

The problem of answering queries to DMKBs with static rules is expressed as follows.

Definition 84 *(Hybrid query answering for updates of DMKBs with static rules).* Let $\diamond$ be a first-order update operator that satisfies (FO8.2) and Ω a complexity class. We say that *hybrid query answering for* $\diamond$ *belongs to* Ω if for every admissible DMKB with static rules $\boldsymbol{K}$ and every finite set of ground generalised literals S, Ω contains the problem of deciding whether $\boldsymbol{K} \models^{\diamond}_{\mathsf{MKNF}} \kappa(S)$.

The following theorem shows that query answering to a DMKB with static rules is a function of the complexity of query answering for the ontology update operator that was used. Specifically, the theorem provides an upper bound, determined by guessing a model J for the ground static rules, then determining whether J corresponds to a model of the DMKB.

Theorem 85. *Let $\diamond$ be a first-order update operator that satisfies (FO8.2) and Ω a complexity class. If query answering for $\diamond$ belongs to Ω, then hybrid query answering for $\diamond$ belongs to* $\mathbf{NP}^{\Omega}$.

Proof (sketch). See Appendix B, page 76. □

6.2.2. Layered DMKBs

Before addressing the problem of answering queries to layered DMKBs, we state the problem of query answering for rule updates. Similarly as in Sections 4 and 5, throughout this subsection we assume that the generalised atom base $\mathcal{B}$ consists of objective literals only.

[23] These complexity results apply to the case where no rule in $\mathcal{P}$ contains a disjunction in its head, a restriction we have assumed throughout this paper.
[24] The definition of admissibility in [77] was not restricted to ground MKNF knowledge bases.

Definition 86 *(Query answering for rule updates).* Let S be a rule update semantics and Ω a complexity class. We say that *query answering for* S *belongs to* Ω if for every admissible DLP $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$, and every finite set of ground literals S, Ω contains the problem of deciding whether for some $J \in [\![\boldsymbol{P}]\!]_{\mathsf{S}}$, $J \models S$.

The problem of answering queries to layered DMKBs is then expressed as follows.

Definition 87 *(Hybrid query answering for updates of layered DMKBs).* Let ⋄ be a first update operator and S a rule update semantics such that both ⋄ and S have the splitting sequence property, conserve the language and respect fact update. Given a complexity class Ω, we say that *hybrid query answering for* ⋄ *and* S belongs to Ω if for every admissible layered DMKB $\boldsymbol{K}$ and every finite set of ground literals S, Ω contains the problem of deciding whether $\boldsymbol{K} \models^{\diamond,\mathsf{S}}_{\mathsf{MKNF}} \kappa(S)$.

In a manner similar to Theorem 85, the following theorem shows that query answering to a layered DMKB is a function of the complexity of query answering for the update operators that were used. Specifically, the theorem provides an upper bound, determined by guessing sub-models of each layer in the splitting sequence of the DMKB, then determining whether these sub-models together correspond to a model of the DMKB.

Theorem 88. *Let ⋄ be a first update operator and* S *a rule update semantics such that both ⋄ and* S *have the splitting sequence property, conserve the language and respect fact update. If query answering for ⋄ belongs to* Ω_1 *and query answering for* S *belongs to* Ω_2*, then hybrid query answering for ⋄ and* S *belongs to* $\mathbf{NP}^{\Omega_1 \cup \Omega_2}$.

Proof (sketch). See Appendix D, page 101. □

Theorems 85 and 88, while abstract, provide upper bounds on computing entailment for admissible DMKBs. Exploring more precise results based on specific ontology and rule update operators, is an avenue for future work.

7. Discussion

In this paper we have taken important first steps towards solving the problem of updates to MKNF knowledge bases, a mature framework for tightly integrated hybrid knowledge representing and reasoning.

We have defined two complementary update semantics. The first semantics (Section 3) applies to DMKBs with static rules, and uses a first-order update operator ⋄ to perform ontology updates. This semantics 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 that can be overridden by ontology updates when necessary. The second semantics applies to layered DMKBs (Section 5), and modularly combines a first-order update operator ⋄ with a rule update semantics S. We demonstrated the use of this semantics in a realistic scenario involving cargo imports. The approach is capable of performing non-trivial updates, automatically resolving conflicts in the expected manner, and propagating new information across the knowledge base. To define the semantics of layered DMKBs we also introduced Abstract Splitting Properties (Section 4), which reformulate the notion of splitting so that it can be parameterised by different logical formalisms; the Abstract Splitting Properties form the basis of the hybrid update semantics defined in Section 5.

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 (cf. Theorem 80). By combining them one thus obtains an integrated hybrid update semantics that can safely handle inputs treated by either of the two semantics.

We also examined the basic theoretical properties of both hybrid update semantics. We showed that they are faithful to ⋄ and S: i.e., the hybrid semantics preserves the behaviour of the underlying update semantics when the hybrid knowledge base contains only ontology axioms or only rules. Similarly, both of the introduced semantics are faithful to the static semantics of MKNF knowledge bases, so that 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. Finally, in Section 6 we have presented basic complexity results on each update semantics.

In defining each semantics, we have made as few assumptions as possible about the properties that guarantee the correctness of our definitions. With respect to update operators, we abstract away from particular instantiations of ⋄ 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 the given assumptions. As an example the assumptions of language conservation and fact update are both needed in Section 5 to guarantee that the layered update semantics is independent of the choice of a splitting; both are basic properties of classical and of rule updates. We have formally shown that they are satisfied for Winslett's update operator and the RD-semantics for rule updates. Similarly it can be shown that they hold for other approaches to classical updates, such as Hegner's semantics [47], Forbus' operator [39], Winslett's standard semantics and its syntax-independent version [50,96], or the modified Winslett's operator [35], as well as by other rule update semantics, such as those introduced in [6,20,30,31,37,66,80,97]. Beyond the assumptions about update operators, few assumptions have been made about the knowledge base, $\mathcal{K}$, that is to be updated. $\mathcal{K}$ may have an ontology constructed from any formalism that is translatable into first-order logic. This includes not only most description logics, but other formalisms such as guarded first-order logics [9]. Of course, update operators for such fragments of first-order logic may not yet have been formulated, but the results in this paper indicate the sufficient conditions if they are to be extended to update MKNF knowledge bases.

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 practical use cases. In particular, the separation of a hybrid knowledge base into distinct ontology and rule layers, as required by the layered update semantics of Section 5, seems to be a natural way of controlling how the different types of knowledge interact, and is in the spirit of recent work on modules for logic programs [12,54].

7.1. Related work

Updates of description logics Recent research has explored various issues pertaining to updating ontologies that are based on description logics. One such issue may be termed the *recovery problem*: given an update operator $\diamond$, and ontologies $\mathcal{O}_1$, $\mathcal{O}_2$ both in a specific DL $\mathcal{D}$, can $\mathcal{O}_1 \diamond \mathcal{O}_2$ be expressed in $\mathcal{D}$? Unfortunately, [10] showed that when $\mathcal{D}$ is at least as expressive as $\mathcal{ALCQI}$ and $\diamond$ is $\diamond_W$, querying $\mathcal{O}_1 \diamond_W \mathcal{O}_2$ is undecidable (and consequently cannot be expressed in $\mathcal{ALCQI}$).

Because of this result, one branch of work has focused on the restriction of updates to information in ABoxes. For DLs that are at least as expressive as $\mathcal{ALC}$, ABox updates may be restricted to those that only allow for updates with assertions about concept names (i.e., assertions about classes to which given individuals belong) [15,36,73]. In these cases, the update semantics used coincides with Winslett's first-order operator $\diamond_W$, 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. An alternate line of work addresses updates of less expressive DLs, such as the DL-Lite [27–29]. Unlike ABox updates for expressive DLs, these update operators ensure that ABox updates do not affect the TBox, treating the TBox the same way that integrity constraints were handled in early belief update semantics [56]. 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. On the positive side, this means that the operator $\diamond_W$ ensures that the TBox is invariant under updates; on the negative side, such an approach may suffer from the same issues that arise when augmenting belief update semantics with integrity constraints [48,49,68].

The ontology update operators just discussed may be termed *syntax-independent* (or model-based), i.e. they satisfy the postulate (B4). However, just as rule update operators are not syntax-independent, ontology update operators may be *formula-based* (cf. [96]). For formula-based operators, no semantic issues must be addressed when lifting operators based on propositional logic to first-order logic or ontologies, and expressibility is not an issue. Accordingly, [21] argued that syntax-independent operators provide inappropriate results when applied to TBoxes and introduced a formula-based operator for TBox updates in DL-Lite. Similarly, [67] have used a formula-based operator to tackle ABox updates to DL-Lite.

Updates of logic programs When updates started to be investigated in the context of Logic Programming, it was only natural to adapt belief update principles and operators to this purpose [4,76]. However, such approaches proved insufficiently expressive, principally because the model-based approach fails to capture the essential relationships between literals encoded in rules [66], and the formula-based approach is too crude as it does not allow rules to be reactivated when reasons for their suppression disappear [97]. Although state-of-the-art approaches to rule updates are guided by the same basic intuitions and aspirations as belief updates, they build upon fundamentally different principles and methods.

Many of them are based on the *causal rejection principle* [5–7,37,66,78]. Alternative approaches to rule updates employ syntactic transformations and other methods, such as abduction [80], prioritisation and preferences [31,97], or dependencies on default assumptions [63,82].

Despite the variety of techniques used in these approaches, certain properties are common to all of them, and seem to be taken as fundamental. First, the stable models assigned to a program after one or more updates are always *supported*: for each true atom p there exists a rule in either the original program or its updates that has p in the head and whose body is satisfied. Second, all mentioned rule update semantics coincide when it comes to *updating sets of facts* by newer facts. Additionally, they need to refer to the *syntactic structure* of a logic program: the individual rules and, in most cases, also the literals in their heads and bodies. This is what makes their full reconciliation with ontology updates so difficult since ontology axioms simply have no heads and bodies.

Despite this impedance mismatch, some are still looking for a unifying framework that could embrace both belief and rule updates. For such a framework, finding an appropriate notion of equivalence is key. In [32,33], AGM based revision was reformulated in the context of Logic Programming in a manner analogous to belief revision in classical propositional logic, and specific revision operators for logic programs were investigated. Central to this novel approach are *SE models* [93] which provide a monotonic semantic characterisation of logic programs that is strictly more expressive than the answer-set semantics. Furthermore, two programs have the same set of SE models if and only if they are strongly equivalent [72], which means that programs $\mathcal{P}$, $\mathcal{Q}$ with the same set of SE models can be modularly replaced by each other with respect to the answer-set semantics, because strong equivalence guarantees that $\mathcal{P} \cup \mathcal{R}$ has the same answer sets as $\mathcal{Q} \cup \mathcal{R}$ for any program $\mathcal{R}$.

In [85,90], a similar path was followed for KM based updates. However, despite the promise that the additional expressiveness of *SE models* could bring, [85,90] also show that *strong equivalence* is not a suitable basis for syntax-independent rule update operators because such operators cannot respect both support and fact update. This can be demonstrated on programs $\mathcal{P} = \{ p., q. \}$ and $\mathcal{Q} = \{ p., q \leftarrow p. \}$ which are strongly equivalent. Due to syntax independence, an update asserting that p is now false ought to lead in both cases to the same stable models. Due to fact update, such an update on $\mathcal{P}$ should naturally lead to a stable model where q is true. But in case of $\mathcal{Q}$ such a stable model would be unsupported. Such examples led to the study of stronger notions of program equivalence in [86,87], and the proposal to view a logic program as the *set of sets of models of its rules* in order to acknowledge rules as the atomic pieces of knowledge and, at the same time, abstract away from unimportant differences between their syntactic forms, focusing on their semantic content.

Within this view of logic programs, updates can be seen to resemble formula-based ontology update operators. In [88] the authors define an abstract framework – dubbed exception driven updates – which is capable of capturing both logic program updates, namely the operator in [66], along with several update operators used for ontology updates, such as the model-based Winslett's operator, or the formula-based WIDTIO and Bold operators [21,29,67,73]. This view is closely related to *base revision operators* [40,46], and provides a promising avenue to further investigate updates of hybrid knowledge bases.

Formalisms for representing hybrid knowledge Multi-context systems [17] provide an alternate formalism for representing hybrid knowledge in which each context represents a logic together with a set of beliefs, and where the contexts are integrated through so-called bridge rules. Multi-context systems are more abstract than MKNF knowledge bases overall and so are more general; however, the relationship of the two formalisms has recently been clarified [60]. From the perspective of the work described in this paper, there is an interesting relationship between the defined class of layered DMKBs and multi-context systems. 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 (cf. Proposition 72). Based on this close relationship, a further study of the update frameworks presented here and the update semantics of managed multi-context systems [18], and more recently of reactive multi-context systems [16,19], and evolving multi-context systems [45] may bring about interesting new insights. Similarly, the relationship remains to be explored with [94], which studies how the first-order belief update postulates (Section 2.5) can be extended to multi-context systems and applied to update operators. Furthermore, the development of similar update semantics under other existing frameworks for hybrid knowledge bases [25,26,38,52,62,64] constitutes another topic of future investigation.

7.2. Future work and extensions

The results of this paper lay the groundwork for practical experimentation. In this context, two implementations exist that support MKNF knowledge bases with stratified programs: [44] which supports $\mathcal{ALCQ}$, and the NoHR plug-in for the Protégé ontology editor which supports $\mathcal{EL}$ [53] and *DL-Lite* [23]. These implementations may serve as starting points to explore the practical behaviour of hybrid updates based on the combination of specific update operators such as those mentioned in Section 7.1, and used in practice to deal with the dynamic aspects of applications which require the expressive power of hybrid knowledge bases, such as normative systems [1,2].

The hybrid update semantics presented in this paper can be further generalised by lifting the various constraints under which they are developed. In case of the semantics for DMKBs with static rules 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 update semantics for layered DMKBs, one issue to study is type of splitting that it is based on. The main limitation is that the current definition splits on *predicate symbols*, while in some cases it would be desirable to allow for a more fine-grained splitting, on the level of ground, and possibly generalised, atoms. More broadly, 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, just as the relationship between dealing with inconsistencies in hybrid knowledge bases through updates and following a paraconsistency approach [55].

Acknowledgements

The authors were partially supported by project ERRO (PTDC/EIA-CCO/121823/2010) and by strategic project NOVA LINCS (PEst/UID/CEC/04516/2013), both funded by Fundação para a Ciência e a Tecnologia.

Appendix A. Proofs: background

In this appendix we introduce a number of formal concepts relating to MKNF knowledge bases and their semantics. These are useful throughout the proofs of propositions and theorems from Sections 3, 4 and 5, presented later in Appendix B, Appendix C and Appendix D, respectively. We also formulate some of the basic properties of these new concepts. For brevity, we leave the proofs of these properties to the reader.

A.1. Properties of first-order and subjective theories

Observation 89 *(Satisfaction of first-order sentences in MKNF structures). Let ϕ be a first-order sentence, $\mathcal{M}, \mathcal{N}$ MKNF interpretations such that $\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{S}$ a set of MKNF interpretations. The following holds:*

$$\begin{array}{rcl} \mathcal{N} \models \mathbf{K}\phi & \textit{implies} & \mathcal{M} \models \mathbf{K}\phi \ ; \\ \forall \mathcal{M}' \in \mathcal{S} : \mathcal{M}' \models \mathbf{K}\phi & \textit{implies} & \left(\bigcup \mathcal{S}\right) \models \mathbf{K}\phi \ . \end{array}$$

Definition 90 *(Subjective MKNF formula and theory, subjective entailment).* An MKNF formula ϕ is *subjective* if all first-order atoms in ϕ occur within the scope of a modal operator. An MKNF theory is *subjective* if all its members are subjective. Furthermore, for any subjective MKNF sentence ϕ, subjective MKNF theory $\mathcal{T}$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ we write

$$\begin{array}{lll} (\mathcal{M}, \mathcal{N}) \models \phi & \text{if and only if} & \exists I \in \mathfrak{I}_{\mathcal{L}} : (I, \mathcal{M}, \mathcal{N}) \models \phi \ , \\ (\mathcal{M}, \mathcal{N}) \models \mathcal{T} & \text{if and only if} & \exists I \in \mathfrak{I}_{\mathcal{L}} : (I, \mathcal{M}, \mathcal{N}) \models \mathcal{T} \ . \end{array}$$

Observation 91 *(Properties of subjective entailment). Let ϕ, ϕ_1, ϕ_2 be subjective MKNF sentences, $\mathcal{T}$ a set of subjective MKNF sentences, π a ground MKNF rule, $I \in \mathfrak{I}_{\mathcal{L}}$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. Then the following holds:*

(1)

$$\begin{array}{lll} (\mathcal{M}, \mathcal{N}) \models \phi & \textit{if and only if} & (I, \mathcal{M}, \mathcal{N}) \models \phi \ ; \\ (\mathcal{M}, \mathcal{N}) \models \mathcal{T} & \textit{if and only if} & (I, \mathcal{M}, \mathcal{N}) \models \mathcal{T} \ ; \end{array}$$

(2) *If $\mathcal{M} \neq \emptyset$, then*

$$\begin{array}{lll} (\mathcal{M}, \mathcal{M}) \models \phi & \textit{if and only if} & \mathcal{M} \models \phi \ ; \\ (\mathcal{M}, \mathcal{M}) \models \mathcal{T} & \textit{if and only if} & \mathcal{M} \models \mathcal{T} \ ; \end{array}$$

(3)

$$\begin{array}{lll} (\mathcal{M}, \mathcal{N}) \models \neg\phi & \textit{if and only if} & (\mathcal{M}, \mathcal{N}) \not\models \phi \ ; \\ (\mathcal{M}, \mathcal{N}) \models \phi_1 \wedge \phi_2 & \textit{if and only if} & (\mathcal{M}, \mathcal{N}) \models \phi_1 \textit{ and } (\mathcal{M}, \mathcal{N}) \models \phi_2 \ ; \\ (\mathcal{M}, \mathcal{N}) \models \phi_1 \supset \phi_2 & \textit{if and only if} & (\mathcal{M}, \mathcal{N}) \models \phi_1 \textit{ implies } (\mathcal{M}, \mathcal{N}) \models \phi_2 \ ; \end{array}$$

(4) *Suppose that $\mathcal{M} \neq \emptyset$ and $\mathcal{N} \neq \emptyset$. Then:*

$$\mathcal{M} \models \kappa(\pi) \quad \textit{if and only if} \quad \mathcal{M} \models \kappa(\mathsf{B}_\pi) \textit{ implies } \mathcal{M} \models \kappa(\mathsf{H}_\pi) \ ;$$

Also, if H_π is a generalised atom, then

$$(\mathcal{M}, \mathcal{N}) \models \kappa(\pi) \quad \textit{iff} \quad \mathcal{M} \models \kappa(\mathsf{B}_\pi^+) \textit{ and } \mathcal{N} \models \kappa(\sim\mathsf{B}_\pi^-) \textit{ implies } \mathcal{M} \models \kappa(\mathsf{H}_\pi) \ ;$$

Also, if H_π is a generalised default literal, then

$$(\mathcal{M}, \mathcal{N}) \models \kappa(\pi) \quad \textit{iff} \quad \mathcal{M} \models \kappa(\mathsf{B}_\pi^+) \textit{ and } \mathcal{N} \models \kappa(\sim\mathsf{B}_\pi^-) \textit{ implies } \mathcal{N} \models \kappa(\mathsf{H}_\pi) \ ;$$

(5) *For any MKNF knowledge base $\mathcal{K}$, 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 $\mathcal{M}' \supsetneq \mathcal{M}$, $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$.*

A.2. Restricted MKNF interpretations

Here we formulate the essential properties of *interpretation restrictions*, defined in Definition 67 on page 61.

Definition 92 *(Interpretation coincidence).* Let A be a set of predicate symbols. For any $I, J \in \mathfrak{I}_{\mathcal{L}}$ 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 $\mathcal{M}^{[A]} = \mathcal{N}^{[A]}$.

Observation 93. *Let A be a set of predicate symbols, $\mathcal{M}, \mathcal{M}', \mathcal{N}, \mathcal{N}' \in \mathfrak{M}$ and $\mathcal{T}$ an MKNF theory such that $A \supseteq \mathsf{pr}(\mathcal{T})$. The following holds:*

(1) *If $\mathcal{M} \subseteq \mathcal{N}$ and $\mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]}$, then $\mathcal{M} \subsetneq \mathcal{N}$.*

(2) *If and* $\mathcal{M}$ *coincides with* $\mathcal{N}$ *on* A*, then*

$$\mathcal{M} \models \mathcal{T} \qquad \text{if and only if} \qquad \mathcal{N} \models \mathcal{T} \enspace .$$

(3) *If* $\mathcal{T}$ *is subjective,* $\mathcal{M}$ *coincides with* $\mathcal{M}'$ *on* A *and* $\mathcal{N}$ *coincides with* $\mathcal{N}'$ *on* A*, then*

$$(\mathcal{M}, \mathcal{N}) \models \mathcal{T} \qquad \text{if and only if} \qquad (\mathcal{M}', \mathcal{N}') \models \mathcal{T} \enspace .$$

A.3. Saturated MKNF interpretations

Here we formulate the essential properties of *saturated MKNF interpretations*, defined in Definition 67 on page 61.

The following concept and its properties, strongly related to saturated interpretations, will be essential for proving our results in Section 5.

Definition 94. Let A be a set of predicate symbols and $\mathcal{M} \in \mathfrak{M}$. We introduce the following notation:

$$\sigma(\mathcal{M}, A) = \left\{ I \in \mathfrak{I}_{\mathcal{L}} \,\middle|\, I^{[A]} \in \mathcal{M}^{[A]} \right\}$$

Also, other properties of saturated MKNF interpretations play an important role.

Observation 95. *Let* A*,* B *be sets of predicate symbols and* $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$*. Then the following holds:*

(1) *If* $\mathcal{M}$ *is saturated relative to* A*, then*

$$\mathcal{M} \subsetneq \mathcal{N} \qquad \text{if and only if} \qquad \mathcal{M} \subseteq \mathcal{N} \text{ and } \mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]} \enspace .$$

(2) *If* $\mathcal{T}$ *is an MKNF theory such that* $A \supseteq \mathrm{pr}(\mathcal{T})$*, and* $\mathcal{M}$ *is an MKNF model of* $\mathcal{T}$*, then* $\mathcal{M}$ *is saturated relative to* A*.*
(3) *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* $\mathcal{M} \subseteq \mathcal{N}$*.*
(4)

$$\sigma(\sigma(\mathcal{M}, A), B) = \sigma(\mathcal{M}, A \cap B) \enspace .$$

(5) *If* $A \subseteq B$*, then*

$$\sigma(\mathcal{M}, B)^{[A]} = \mathcal{M}^{[A]} \enspace .$$

(6) *If* $A \subseteq B$ *and* $\mathcal{M}$ *is saturated relative to* A*, then it is also saturated relative to* B*.*
(7) *If* A*,* B *are disjoint, then*

$$\sigma(\mathcal{M}, A)^{[B]} = \mathfrak{I}_{\mathcal{L}}^{[B]} \enspace .$$

Note particularly that Observation 95(1) is a strengthened version of Observation 93(1), with the implication replaced by an equivalence, that holds only for saturated MKNF interpretations.

A.4. Semi-saturated MKNF interpretations

Another important class of MKNF interpretations for which a slightly modified version of Observation 93(1) holds are *semi-saturated* interpretations. After defining them formally, we list some of their most essential properties.

Definition 96 *(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 $I \in \mathfrak{I}_{\mathcal{L}}$,

$$I^{[A]} \in \mathcal{M}^{[A]} \land I^{[\mathcal{P} \setminus A]} \in \mathcal{M}^{[\mathcal{P} \setminus A]} \qquad \text{implies} \qquad I \in \mathcal{M} \enspace .$$

Observation 97. *Let* A *be a set of predicate symbols and* $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$*. Then the following holds:*

(1) *If* $\mathcal{M}$ *is semi-saturated relative to* A*, then*

$$\mathcal{M} \subsetneq \mathcal{N} \qquad \text{if and only if} \qquad \mathcal{M} \subseteq \mathcal{N} \text{ and either } \mathcal{M}^{[A]} \subsetneq \mathcal{N}^{[A]} \text{ or } \mathcal{M}^{[\mathcal{P} \setminus A]} \subsetneq \mathcal{N}^{[\mathcal{P} \setminus A]} \enspace .$$

(2) *There exists the greatest $\mathcal{M}_0 \in \mathfrak{M}$ that coincides with $\mathcal{M}$ on A and with $\mathcal{N}$ on $\mathfrak{P} \setminus A$. Furthermore, $\mathcal{M}_0$ is semi-saturated relative to A and $\mathcal{M} \cap \mathcal{N} \subseteq \mathcal{M}_0$.*
Also, if $\mathcal{M}$ is saturated relative to A and $\mathcal{N}$ relative to $\mathfrak{P} \setminus A$, then $\mathcal{M}_0 = \mathcal{M} \cap \mathcal{N}$.
Finally, if $\mathcal{M} \neq \emptyset$ and $\mathcal{N} \neq \emptyset$, then $\mathcal{M}_0 \neq \emptyset$.

A.5. Sequence-saturated MKNF interpretations

The properties 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 Section 4.2.

Definition 98 *(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 $\bigcup_{\alpha<\mu} A_\alpha = \mathfrak{P}$.

Definition 99 *(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 $I \in \mathfrak{I}_{\mathcal{L}}$,

$$\forall \alpha < \mu : I^{[A_\alpha]} \in \mathcal{M}^{[A_\alpha]} \qquad \text{implies} \qquad I \in \mathcal{M} \enspace .$$

Observation 100. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence and $\mathcal{M} \in \mathfrak{M}$. Then the following holds:*

(1) *These 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.*

(2) *Let $\langle \mathcal{M}_\alpha \rangle_{\alpha<\mu}$ be a sequence of members of $\mathfrak{M}$ such that for all $\alpha < \mu$, $\mathcal{M}_\alpha$ is saturated relative to A_α, and $\mathcal{M} = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha$. Then:*
  (i) *$\mathcal{M} = \emptyset$ if and only if for some $\alpha < \mu$, $\mathcal{M}_\alpha = \emptyset$.*
  (ii) *If $\mathcal{M} \neq \emptyset$, then for all $\alpha < \mu$, $\mathcal{M}_\alpha = \sigma(\mathcal{M}, A_\alpha)$.*

Appendix B. Proofs: dynamic MKNF knowledge bases with static rules

In the following we present proofs of results from Section 3, implicitly working under the same assumptions as those imposed in that section. That is, we assume that all MKNF rules are ground and do not allow for rules with default negation in their heads. In the proofs we also implicitly use many of the results presented in Appendix A.

B.1. Static consequence operator

B.1.1. Static consequence operator for positive MKNF programs

Lemma 101 *(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 $\mathcal{N} \models \kappa(\mathsf{B}_\pi)$ and $\mathsf{H}_\pi = \xi$. It follows from Observation 89 that $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$ and, thus, $\xi \in T_{\mathcal{P}}(\mathcal{M})$. □

Proposition 21 *(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 101 that $T_{\mathcal{P}}(\mathcal{M}) \supseteq T_{\mathcal{P}}(\mathcal{N})$ and, consequently,

$$T_{\mathcal{K}}(\mathcal{M}) = [\![T_{\mathcal{P}}(\mathcal{M}) \cup \kappa(\mathcal{O})]\!] \subseteq [\![T_{\mathcal{P}}(\mathcal{N}) \cup \kappa(\mathcal{O})]\!] = T_{\mathcal{K}}(\mathcal{N}) \enspace . \qquad \square$$

Definition 102. We say that an MKNF interpretation $\mathcal{M}$ is an *S5 model* of an MKNF theory $\mathcal{T}$ if $\mathcal{M} \models \mathcal{T}$. Similarly, $\mathcal{M}$ is an *S5 model* of an MKNF knowledge base $\mathcal{K}$ if it is an S5 model of $\kappa(\mathcal{K})$.[25]

Lemma 103. *If $\mathcal{M}$ is an S5 model of a first-order theory $\mathcal{T}$, then $\mathcal{M} \subseteq [\![\mathcal{T}]\!]$.*

[25] Such models are so named because an S5 model $\mathcal{M}$ for a closed formula ϕ is obtained by interpreting ϕ as a first-order model formula in the modal logic S5 where **not** is taken to be a shortcut for $\neg\mathbf{K}$ [77].

Proof. If $\mathcal{M}$ is an S5 model of $\mathcal{T}$, then $\mathcal{M} \models \mathcal{T}$ and it follows that $\mathcal{M}$ only contains first-order models of $\mathcal{T}$. Thus, by the definition of $[\![.]\!]$, $\mathcal{M}$ is a subset of $[\![T]\!]$. □

Lemma 104. *For any first-order theory $\mathcal{T}$, either $\mathcal{T}$ has no S5 model, or $[\![\mathcal{T}]\!]$ is the greatest S5 model of $\mathcal{T}$.*

Proof. Consider two cases:

- a) If $[\![\mathcal{T}]\!]$ is empty, then it follows from Lemma 103 that $\mathcal{T}$ has no S5 model (every S5 model must be an MKNF interpretation, thus non-empty).
- b) if $[\![\mathcal{T}]\!]$ is not empty, then from Lemma 103 and from $[\![\mathcal{T}]\!] \models \mathcal{T}$ we immediately conclude that $[\![\mathcal{T}]\!]$ is the greatest S5 model of $\mathcal{T}$. □

Lemma 105. *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 $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$ it holds that $\mathcal{M} \models \mathsf{H}_\pi$. In other words, $\mathcal{M}$ is an S5 model of $T_{\mathcal{P}}(\mathcal{M}) \cup \kappa(\mathcal{O})$. By Lemma 104, $T_{\mathcal{K}}(\mathcal{M}) = [\![T_{\mathcal{P}}(\mathcal{M})) \cup \kappa(\mathcal{O})]\!]$ 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 $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$, it holds that $\mathcal{M} \models \kappa(\mathsf{H}_\pi)$. Consequently, $\mathcal{M}$ is an S5 model of $\mathcal{K}$. □

Lemma 106. *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 $\mathcal{K}$.

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 Observation 89 that $\mathcal{M} \models \mathbf{K}\kappa(\phi)$.

The other possibility is that ϕ is a sentence of the form $\kappa(\mathsf{B}_\pi) \supset \kappa(\mathsf{H}_\pi)$ for some $\pi \in \mathcal{P}$ (cf. Definition 12). Suppose that $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$. As π is positive, it follows from Observation 89 that $\mathcal{N} \models \kappa(\mathsf{B}_\pi)$ for every S5 model $\mathcal{N}$ of $\mathcal{K}$ and, thus, $\mathcal{N} \models \kappa(\mathsf{H}_\pi)$. Hence, by Observation 89, $\mathcal{M} \models \kappa(\mathsf{H}_\pi)$.

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 (Definitions 8 and 9) 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 23 *(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 21, 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 105, $\mathcal{S}$ 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 106, it coincides with the unique MKNF model of $\mathcal{K}$. □

B.1.2. Static consequence operator for MKNF programs with default negation

Lemma 107. *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 $\pi' = (\mathsf{H}_\pi \leftarrow \mathsf{B}^+_\pi.)$ from $\mathcal{P}^{\mathcal{M}}$ for some $\pi \in \mathcal{P}$ with $\mathcal{M} \models \kappa(\sim\mathsf{B}^-_\pi)$. Then $\kappa(\mathcal{K}^{\mathcal{M}})$ contains the sentence $\kappa(\pi')$ of the form $\kappa(\mathsf{B}^+_\pi) \supset \kappa(\mathsf{H}_\pi)$. If $\mathcal{M} \models \kappa(\mathsf{B}^+_\pi)$, then it follows that $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$ and since $\mathcal{M} \models \kappa(\pi)$, it follows that $\mathcal{M} \models \kappa(\mathsf{H}_\pi)$. 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 $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$, then $\mathcal{P}^{\mathcal{M}}$ contains the rule $\pi' = (\mathsf{H}_\pi \leftarrow \mathsf{B}^+_\pi.)$ and since $\mathcal{M} \models \kappa(\pi')$, it follows that $\mathcal{M} \models \kappa(\mathsf{H}_\pi)$. Hence, $\mathcal{M} \models \kappa(\pi)$. □

Proposition 26 *(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 107.

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 $\mathcal{K}^{\mathcal{M}}$ (cf. Lemma 106).

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 $\kappa(\mathsf{B}_\pi) \supset \kappa(\mathsf{H}_\pi)$ for some rule $\pi \in \mathcal{P}$ and the following must hold:

$$(\mathcal{M}', \mathcal{M}) \models \kappa(\sim\mathsf{B}_\pi^-) \wedge (\mathcal{M}', \mathcal{M}) \models \kappa(\mathsf{B}_\pi^+) \wedge (\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathsf{H}_\pi)$$

which is equivalent to

$$\mathcal{M} \models \kappa(\sim\mathsf{B}_\pi^-) \wedge \mathcal{M}' \models \kappa(\mathsf{B}_\pi^+) \wedge \mathcal{M}' \not\models \kappa(\mathsf{H}_\pi) \ .$$

However, this is in conflict with $\mathcal{M}'$ being an S5 model of $\mathcal{K}^{\mathcal{M}}$ since the sentence

$$\kappa(\mathsf{B}_\pi^+) \supset \kappa(\mathsf{H}_\pi)$$

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 107 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 $\pi' = (\mathsf{H}_\pi \leftarrow \mathsf{B}_\pi^+.) \in \mathcal{P}^{\mathcal{M}}$ such that $\mathcal{M}' \not\models \kappa(\pi')$, i.e.

$$\mathcal{M} \models \kappa(\sim\mathsf{B}_\pi^-) \wedge \mathcal{M}' \models \kappa(\mathsf{B}_\pi^+) \wedge \mathcal{M}' \not\models \mathsf{H}_\pi \ .$$

This is equivalent to

$$(\mathcal{M}', \mathcal{M}) \models \kappa(\sim\mathsf{B}_\pi^-) \wedge (\mathcal{M}', \mathcal{M}) \models \kappa(\mathsf{B}_\pi^+) \wedge (\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathsf{H}_\pi)$$

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 30 *(Monotonicity of $T_{\boldsymbol{K}}^{\diamond}$). Let $\diamond$ be a first-order update operator and $\boldsymbol{K}$ a positive DMKB with static rules. If $\diamond$ satisfies (FO8.2), then $T_{\boldsymbol{K}}^{\diamond}$ 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 $T_{\boldsymbol{K}}^{\diamond}(\mathcal{M}) \subseteq T_{\boldsymbol{K}}^{\diamond}(\mathcal{N})$.

By Lemma 101 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

$$\left(\left(T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0)\right) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}\right) \models \left(\left(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0)\right) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}\right) \ .$$

Consequently, $T_{\boldsymbol{K}}^{\diamond}(\mathcal{M}) \subseteq T_{\boldsymbol{K}}^{\diamond}(\mathcal{N})$. □

B.3. Properties and relations

Theorem 37 *(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 23 and 26, 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 $T_{\langle \mathcal{K} \rangle}^{\diamond}$. □

Theorem 38 *(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 39 *(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 40 *(Immunity to tautological updates). Suppose that $\diamond$ satisfies (FO2.$\top$) 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

$$T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = [\![\mathcal{T} \diamond \mathcal{O}_j \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \qquad \text{and} \qquad T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N}) = [\![\mathcal{T} \diamond \mathcal{O}_{j+1} \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \enspace .$$

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 41 *(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

$$T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = [\![\left(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0)\right) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \enspace ,$$
$$T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N}) = [\![\left(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}'_0)\right) \diamond \mathcal{O}'_1 \diamond \cdots \diamond \mathcal{O}'_{n-1}]\!] \enspace .$$

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

$$\left(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}_0)\right) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1} \equiv \left(T_{\mathcal{P}_0}(\mathcal{N}) \cup \kappa(\mathcal{O}'_0)\right) \diamond \mathcal{O}'_1 \cdots \diamond \mathcal{O}'_{n-1} \enspace .$$

Consequently, $T^{\diamond}_{\boldsymbol{K}^{\mathcal{M}}}(\mathcal{N}) = T^{\diamond}_{(\boldsymbol{K}')^{\mathcal{M}}}(\mathcal{N})$. □

B.4. Complexity of updating DMKBs with static rules

The proof of Theorem 85 requires a counterpart of the updating consequence operator of Definition 28 that, for a DMKB $\boldsymbol{K}$ and a set of head atoms in the static rules of $\boldsymbol{K}$, produces a new set of entailed head atoms as follows.

Definition 108. 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. Also, let HA' be the set of all (generalised) head atoms in the rules of $\mathcal{P}_i$. $\text{T-syn}^{\diamond}_{\boldsymbol{K}}$ is defined for all $\mathcal{M} \in \mathfrak{M}$ and $\mathsf{HA}_\top \subseteq \mathsf{HA}'$ as follows:

$$\text{T-}\overset{\diamond}{\underset{\boldsymbol{K}}{\text{syn}}}(\mathcal{M}) = \left\{ \mathsf{H}_\pi \;\middle|\; \pi \in \mathcal{P} \wedge [\![(\kappa(\mathsf{HA}_\top) \cup \kappa(\mathcal{O}_0)) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}]\!] \models \kappa(\mathsf{B}_\pi) \right\} \enspace .$$

Theorem 85. *Let $\diamond$ be a first-order update operator that satisfies (FO8.2) and Ω a complexity class. If query answering for $\diamond$ belongs to Ω, then hybrid query answering for $\diamond$ belongs to $\mathbf{NP}^{\Omega}$.*

Proof (sketch). Take a DMKB with static rules $\boldsymbol{K} = \langle(\mathcal{O}_0, \mathcal{P}), (\mathcal{O}_1, \emptyset), \dots, (\mathcal{O}_{n-1}, \emptyset)\rangle$ and a finite set of ground literals S. We need to prove that the problem of deciding whether

$$\boldsymbol{K} \models^{\diamond}_{\mathsf{MKNF}} \kappa(S)$$

belongs in $\mathbf{NP}^{\Omega}$. In other words, we need to decide whether for some $\diamond$-dynamic MKNF model $\mathcal{M}$ of $\boldsymbol{K}$, $\mathcal{M} \models \kappa(S)$.

Let HA' be the (finite) set of generalised atoms that appear in heads of $\mathcal{P}$. We guess $\mathsf{HA}_\top \subseteq HA'$, then check that

$$\mathcal{M} = [\![(\kappa(\mathsf{HA}_\top) \cup \mathcal{O}_0) \diamond \mathcal{O}_1 \diamond \cdots \diamond \mathcal{O}_{n-1}]\!]$$

is an MKNF model of $\boldsymbol{K}$. In other words, we guess the set $\mathsf{HA}_\top$ of generalised atoms in the heads of rules in $\mathcal{P}$ that are true in $\mathcal{M}$.

The rest can be verified in deterministic polynomial time using our Ω oracle as follows. First, we form the program reduct $\mathcal{P}^{\mathcal{M}}$ by using the oracle to see which default body literals are entailed by $\mathcal{M}$ and which ones are not. Then, we check that $\mathsf{HA}_\top$ is the least fixed point of $\text{T-syn}^{\diamond}_{\boldsymbol{K},\mathsf{HA}_\top}$. Since $\boldsymbol{K}$ is finite and ground, HA' is also finite and ground and is part of the input string of the decision problem. Accordingly, obtaining the fixed point can be achieved by using the oracle polynomially many times. Finally, we check that $\mathcal{M} \models \kappa(\phi)$ by one additional call to the oracle. □

Appendix C. Proofs: semantics with splitting properties

C.1. MKNF knowledge bases

Remark 109. Note that whenever U is a splitting set for an MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, the following holds:

$$\mathsf{pr}(b_U(\mathcal{O})) \subseteq U \enspace , \qquad \mathsf{pr}(b_U(\mathcal{P})) \subseteq U \enspace , \qquad \mathsf{pr}(b_U(\mathcal{K})) \subseteq U \enspace , \qquad \mathsf{pr}(t_U(\mathcal{O})) \subseteq \mathfrak{P} \setminus U \enspace .$$

Also note that the heads of rules in $t_U(\mathcal{P})$ contain only predicate symbols from $\mathfrak{P} \setminus U$ while their bodies may also contain predicate symbols from U. However, for any $\mathcal{X} \in \mathfrak{M}$, the reducts $e_U(\mathcal{P}, \mathcal{X})$ and $e_U(\mathcal{K}, \mathcal{X})$ contain only predicate symbols from $\mathfrak{P} \setminus U$:

$$\mathsf{pr}(e_U(\mathcal{P}, \mathcal{X})) \subseteq \mathfrak{P} \setminus U \enspace , \qquad \mathsf{pr}(e_U(\mathcal{K}, \mathcal{X})) \subseteq \mathfrak{P} \setminus U \enspace .$$

These basic observations will be used in the following proofs without further notice or reference.

C.1.1. Splitting set theorem

Definition 110 *(Generalised splitting set reduct).* Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be an MKNF knowledge base, $U \subseteq \mathfrak{P}$ a set of predicate symbols and $\mathcal{X}, \mathcal{X}' \in \mathfrak{M}$. We define the *reduct of $\mathcal{P}$ relative to U and $(\mathcal{X}', \mathcal{X})$* as

$$e_U(\mathcal{P}, (\mathcal{X}', \mathcal{X})) = \left\{ \mathsf{H}_\pi \leftarrow \{ L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq \mathfrak{P} \setminus U \}. \mid \pi \in t_U(\mathcal{P}) \wedge (\mathcal{X}', \mathcal{X}) \models \kappa(\{ L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq U \}) \right\} \enspace .$$

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 111. *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. $\mathcal{F}^{[\mathfrak{P} \setminus U]} = \mathcal{D}^{[\mathfrak{P} \setminus U]}$ *and* $\mathcal{F}'^{[\mathfrak{P} \setminus U]} = \mathcal{D}'^{[\mathfrak{P} \setminus U]}$;
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 Observation 93(3) 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 Observation 93(3) 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 Observation 91(3), in order to prove that $(\mathcal{F}', \mathcal{F}) \models \kappa(\sigma)$, we can instead show that $(\mathcal{F}', \mathcal{F}) \models \kappa(\mathsf{H}_\sigma)$ given the assumption that $(\mathcal{F}', \mathcal{F}) \models \kappa(\mathsf{B}_\sigma)$. This assumption together with Observation 93(3) implies that $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathsf{B}_\sigma)$ and, by the definition of $e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))$, there must be some rule $\pi \in \mathcal{P}$ such that $\mathsf{H}_\pi = \mathsf{H}_\sigma$ and $(\mathcal{G}', \mathcal{G}) \models \kappa(\mathsf{B}_\pi \setminus \mathsf{B}_\sigma)$. From the last property and Observation 93(3) we obtain $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathsf{B}_\pi \setminus \mathsf{B}_\sigma)$. Thus, $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathsf{B}_\pi)$ and since $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K})$ and $\kappa(\mathcal{K})$ contains $\kappa(\pi)$, we conclude that $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathsf{H}_\pi)$. Consequently, since $\mathsf{H}_\pi = \mathsf{H}_\sigma$, by Observation 93(3) we obtain $(\mathcal{F}', \mathcal{F}) \models \kappa(\mathsf{H}_\sigma)$ 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 112. *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. $\mathcal{F}^{[\mathfrak{P} \setminus U]} = \mathcal{D}^{[\mathfrak{P} \setminus U]}$ *and* $\mathcal{F}'^{[\mathfrak{P} \setminus U]} = \mathcal{D}'^{[\mathfrak{P} \setminus U]}$*;*
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 Observation 93(3) 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 Observation 93(3) we conclude that $(\mathcal{D}', \mathcal{D}) \models \kappa(\pi)$. If $\pi \in t_U(\mathcal{P})$, then assuming that $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathsf{B}_\pi)$ we need to prove that $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathsf{H}_\pi)$. It follows by Observation 93(3) that

$$(\mathcal{G}', \mathcal{G}) \models \kappa(\{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq U \,\})$$

Consequently, $e_U(\mathcal{P}, (\mathcal{G}', \mathcal{G}))$ contains a rule σ such that $\mathsf{H}_\sigma = \mathsf{H}_\pi$ and

$$\mathsf{B}_\sigma = \{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq \mathfrak{P} \setminus U \,\} \enspace .$$

Thus, by Observation 93(3), $(\mathcal{F}', \mathcal{F}) \models \kappa(\mathsf{B}_\sigma)$ and from $(\mathcal{F}', \mathcal{F}) \models e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))$ it follows that $(\mathcal{F}', \mathcal{F}) \models \kappa(\mathsf{H}_\sigma)$. By another application of Observation 93(3) we obtain that $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathsf{H}_\sigma)$ and since $\mathsf{H}_\sigma = \mathsf{H}_\pi$, it follows that $(\mathcal{D}', \mathcal{D}) \models \kappa(\pi)$. Consequently, $(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K})$. □

Proposition 113. *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. $\mathcal{F}^{[\mathfrak{P} \setminus U]} = \mathcal{D}^{[\mathfrak{P} \setminus U]}$ *and* $\mathcal{F}'^{[\mathfrak{P} \setminus U]} = \mathcal{D}'^{[\mathfrak{P} \setminus U]}$*;*
3. $\mathcal{G}^{[U]} = \mathcal{D}^{[U]}$ *and* $\mathcal{G}'^{[U]} = \mathcal{D}'^{[U]}$*.*

Then,

$$(\mathcal{D}', \mathcal{D}) \models \kappa(\mathcal{K}) \qquad \textit{if and only if} \qquad (\mathcal{E}', \mathcal{E}) \models \kappa(b_U(\mathcal{K})) \wedge (\mathcal{F}', \mathcal{F}) \models \kappa(e_U(\mathcal{K}, (\mathcal{G}', \mathcal{G}))) \enspace .$$

Proof. Follows by Lemmas 111 and 112. □

Corollary 114. *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,*

$$\mathcal{M} \models \kappa(\mathcal{K}) \qquad \textit{if and only if} \qquad \mathcal{M} \models \kappa(b_U(\mathcal{K})) \wedge \mathcal{M} \models \kappa(e_U(\mathcal{K}, \mathcal{X})) \enspace .$$

Proof. Proposition 113 for $\mathcal{D} = \mathcal{D}' = \mathcal{E} = \mathcal{E}' = \mathcal{F} = \mathcal{F}' = \mathcal{M}$ and $\mathcal{G} = \mathcal{G}' = \mathcal{X}$ implies that

$$(\mathcal{M}, \mathcal{M}) \models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{M}, \mathcal{M}) \models \kappa(b_U(\mathcal{K})) \wedge (\mathcal{M}, \mathcal{M}) \models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

The claim now follows from Observation 91(2). □

Corollary 115. *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,*

$$\mathcal{M} \models \kappa(\mathcal{K}) \quad \textit{if and only if} \quad \mathcal{X} \models \kappa(b_U(\mathcal{K})) \wedge \mathcal{Y} \models \kappa(e_U(\mathcal{K}, \mathcal{X})) \ .$$

Proof. Proposition 113 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

$$(\mathcal{M}, \mathcal{M}) \models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{X}, \mathcal{X}) \models \kappa(b_U(\mathcal{K})) \wedge (\mathcal{Y}, \mathcal{Y}) \models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

The claim now follows from Observation 91(2). □

Corollary 116. *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,*

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \textit{implies} \quad (\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \ .$$

Proof. Proposition 113 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

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \vee (\mathcal{M}, \mathcal{M}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

Furthermore, from Corollary 114 we know that $\mathcal{M} \models \kappa(e_U(\mathcal{K}, \mathcal{X}))$ is satisfied because $\mathcal{M} \models \kappa(\mathcal{K})$. Hence, by Observation 91(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 117. *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,*

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \textit{implies} \quad (\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X})) \ .$$

Proof. Proposition 113 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

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{M}, \mathcal{M}) \not\models \kappa(b_U(\mathcal{K})) \vee (\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

Furthermore, from Corollary 114 we know that $\mathcal{M} \models \kappa(b_U(\mathcal{K}))$ is satisfied because $\mathcal{M} \models \kappa(\mathcal{K})$. Hence, by Observation 91(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 118. *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,*

$$(\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \quad \textit{implies} \quad (\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \ .$$

Proof. Proposition 113 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

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K})) \vee (\mathcal{M}', \mathcal{M}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{M}', \mathcal{M}))) \ .$$

The claim of this corollary follows directly from this equivalence. □

Corollary 119. *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,*

$$(\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X})) \quad \textit{implies} \quad (\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \ .$$

Proof. Proposition 113 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

$$(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K}) \quad \text{if and only if} \quad (\mathcal{M}, \mathcal{M}) \not\models \kappa(b_U(\mathcal{K})) \vee (\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

The claim of this corollary follows from this equivalence by Observation 91(2). □

Proposition 120. *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 Observation 95(3) we know that $\mathcal{M} \subseteq \mathcal{X}$ and that $\mathcal{X}$ is saturated relative to U. By Observation 91(5), $\mathcal{X}$ 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 115. To verify the latter, pick some $\mathcal{X}' \supsetneq \mathcal{X}$. By Observation 97(2) 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 Observation 95(1) 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 Observation 93(1), 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 116 to conclude that $(\mathcal{X}', \mathcal{X}) \not\models \kappa(b_U(\mathcal{K}))$, which is the desired conclusion. □

Proposition 121. *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 Observation 95(3) we know that $\mathcal{M} \subseteq \mathcal{Y}$ and that $\mathcal{Y}$ is saturated relative to $\mathcal{P} \setminus U$. By Observation 91(5), $\mathcal{Y}$ 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 115. To verify the latter, pick some $\mathcal{Y}' \supsetneq \mathcal{Y}$. By Observation 97(2) 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 Observation 95(1) 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 Observation 93(1), 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 117 to conclude that $(\mathcal{Y}', \mathcal{Y}) \not\models \kappa(e_U(\mathcal{K}, \mathcal{X}))$, which is the desired conclusion. □

Proposition 122. *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 Observation 95(2), $\mathcal{X}$ 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 Observation 97(2), $\mathcal{M}$ 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 115, $\mathcal{M} \models \kappa(\mathcal{K})$.

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 Observation 97(1) one of the following cases must occur:

a) If $\mathcal{M}'^{[U]} \supsetneq \mathcal{M}^{[U]}$, then

$$\mathcal{X}'^{[U]} = \mathcal{X}^{[U]} \cup \mathcal{M}'^{[U]} = \mathcal{M}'^{[U]} \supsetneq \mathcal{M}^{[U]} = \mathcal{X}^{[U]} \ ,$$

so Observation 93(1) 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 118 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

$$\mathcal{Y}'^{[\mathcal{P}\setminus U]} = \mathcal{Y}^{[\mathcal{P}\setminus U]} \cup \mathcal{M}'^{[\mathcal{P}\setminus U]} = \mathcal{M}'^{[\mathcal{P}\setminus U]} \supsetneq \mathcal{M}^{[\mathcal{P}\setminus U]} = \mathcal{Y}^{[\mathcal{P}\setminus U]} \ ,$$

so Observation 93(1) 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 119 we obtain $(\mathcal{M}', \mathcal{M}) \not\models \kappa(\mathcal{K})$ as desired. □

Theorem 123 *(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 120 we obtain that $\mathcal{X} = \sigma(\mathcal{M}, U)$ is an MKNF model of $b_U(\mathcal{K})$ and by Proposition 121 that $\mathcal{Y} = \sigma(\mathcal{M}, \mathcal{P} \setminus U)$ is an MKNF model of $e_U(\mathcal{K}, \mathcal{X})$. Furthermore, by Observation 95(3), $\mathcal{X}^{[U]} = \mathcal{M}^{[U]}$ 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 Observation 97(2):

$$(\mathcal{X} \cap \mathcal{Y})^{[U]} = \mathcal{X}^{[U]} = \mathcal{M}^{[U]} \ ,$$
$$(\mathcal{X} \cap \mathcal{Y})^{[\mathcal{P} \setminus U]} = \mathcal{Y}^{[\mathcal{P} \setminus U]} = \mathcal{M}^{[\mathcal{P} \setminus U]} \ .$$

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 113 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

$$(\mathcal{M}, \mathcal{M}) \not\models \kappa(b_U(\mathcal{K})) \vee (\mathcal{M}, \mathcal{M}) \not\models \kappa(e_U(\mathcal{K}, (\mathcal{X}, \mathcal{X}))) \ .$$

However, Observation 91(2) and Corollary 114 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 122. □

Corollary 124. *Let U be a splitting set for an MKNF knowledge base $\mathcal{K}$ and $\mathcal{M} \in \mathfrak{M}$. 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 123 and of Observation 97(2). □

C.1.2. Splitting sequence theorem

Remark 125. 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,

$$\mathrm{pr}(b_{U_0}(\mathcal{K})) \subseteq U_0 \ ,$$
$$\mathrm{pr}\left(e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \textstyle\bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right)\right) \subseteq U_{\alpha+1} \setminus U_\alpha \text{ whenever } \alpha + 1 < \mu$$

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, $\mathcal{X}_\alpha = \mathfrak{I}_\mathcal{L}$, 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 [92].

Lemma 126. *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 $\mathfrak{M}$ 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 $\mathrm{pr}(p) \subseteq U$,

$$I \models p \qquad \text{if and only if} \qquad J \models p \ .$$

We need to show that I belongs to $\mathcal{X}$. Take some $\beta < \mu$ and some atom q such that $\mathrm{pr}(q) \subseteq U_\beta$. 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 127 *(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 128. *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 $\beta < \alpha < \mu$:*

- $\bigcap_{\gamma\le\beta} \mathcal{X}_\gamma$ *is saturated relative to* U_β*;*
- $\bigcap_{\beta<\gamma\le\alpha} \mathcal{X}_\gamma$ *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 126 we obtain that $\bigcap_{\gamma\le\beta} \mathcal{X}_\gamma$ is saturated relative to

$$\bigcup_{\gamma\le\beta} A_\gamma = A_0 \cup \bigcup_{\gamma<\beta} A_{\gamma+1} = U_0 \cup \bigcup_{\gamma<\beta} U_{\gamma+1} \setminus U_\gamma = U_\beta \ .$$

The same lemma implies that $\bigcap_{\beta<\gamma\le\alpha} \mathcal{X}_\gamma$ must be saturated relative to

$$\bigcup_{\beta<\gamma\le\alpha} A_\gamma = \bigcup_{\beta\le\gamma<\alpha} A_{\gamma+1} = \bigcup_{\beta\le\gamma<\gamma+1\le\alpha} U_{\gamma+1} \setminus U_\gamma = U_\alpha \setminus U_\beta$$

and that $\bigcap_{\alpha<\gamma<\mu} \mathcal{X}_\gamma$ must be saturated relative to

$$\bigcup_{\alpha<\gamma<\mu} A_\gamma = \bigcup_{\alpha\le\gamma<\gamma+1<\mu} A_{\gamma+1} = \bigcup_{\alpha\le\gamma<\gamma+1<\mu} U_{\gamma+1} \setminus U_\gamma = \mathcal{P} \setminus U_\alpha \ . \qquad \square$$

Lemma 129. *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*

$$\bigcap_{\beta\le\alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha) \ .$$

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, $\bigcap_{\beta\le\alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$. We immediately obtain:

$$\begin{aligned}\bigcap_{\beta\le\alpha+1} \mathcal{X}_\beta &= \mathcal{X}_{\alpha+1} \cap \bigcap_{\beta\le\alpha} \mathcal{X}_\beta = \mathcal{X}_{\alpha+1} \cap \sigma(\mathcal{M}, U_\alpha) \\ &= \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha) \cap \sigma(\mathcal{M}, U_\alpha) \ .\end{aligned}$$

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 124 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 124 we obtain that

$$\sigma(\mathcal{M}, U_{\alpha+1}) = \mathcal{N} = \sigma(\mathcal{N}, U_\alpha) \cap \sigma(\mathcal{N}, \mathcal{P} \setminus U_\alpha) \ . \tag{C.5}$$

Moreover, Observation 95(4) 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

$$\begin{aligned}\sigma(\mathcal{N}, \mathcal{P} \setminus U_\alpha) &= \sigma(\sigma(\mathcal{M}, U_{\alpha+1}), \mathcal{P} \setminus U_\alpha) = \sigma(\mathcal{M}, U_{\alpha+1} \cap (\mathcal{P} \setminus U_\alpha)) \\ &= \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha) \ .\end{aligned} \tag{C.7}$$

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 $\bigcap_{\gamma\le\beta} \mathcal{X}_\gamma = \sigma(\mathcal{M}, U_\beta)$. First note that

$$\begin{aligned}\bigcap_{\beta\le\alpha} \mathcal{X}_\beta = \mathcal{X}_\alpha \cap \bigcap_{\beta<\alpha} \mathcal{X}_\beta = \mathcal{I}_\mathcal{L} \cap \bigcap_{\beta<\alpha} \bigcap_{\gamma\le\beta} \mathcal{X}_\gamma &= \bigcap_{\beta<\alpha} \sigma(\mathcal{M}, U_\beta) \\ &= \bigcap_{\beta<\alpha} \left\{ I \in \mathcal{I}_\mathcal{L} \;\middle|\; \exists J \in \mathcal{M} : J^{[U_\beta]} = I^{[U_\beta]} \right\} \\ &= \left\{ I \in \mathcal{I}_\mathcal{L} \;\middle|\; \forall \beta < \alpha\ \exists J \in \mathcal{M} : J^{[U_\beta]} = I^{[U_\beta]} \right\}\end{aligned}$$

and also that

$$\sigma(\mathcal{M}, U_\alpha) = \sigma\left(\mathcal{M}, \bigcup_{\beta<\alpha} U_\beta\right) = \left\{ I \in \mathcal{I}_\mathcal{L} \mid \exists J \in \mathcal{M} : J^{\left[\bigcup_{\beta<\alpha} U_\beta\right]} = I^{\left[\bigcup_{\beta<\alpha} U_\beta\right]} \right\} .$$

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 $J^{\left[\bigcup_{\beta<\alpha} U_\beta\right]} = I^{\left[\bigcup_{\beta<\alpha} U_\beta\right]}$, hence for any $\beta_0 < \alpha$ and any atom p such that $\mathrm{pr}(p) \subseteq U_{\beta_0} \subseteq \bigcup_{\beta<\alpha} U_\beta$ we obtain

$$J \models p \qquad \text{if and only if} \qquad I \models p \ ,$$

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 124 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

$$(\mathcal{Y}, \sigma(\mathcal{M}, U_\alpha)) \not\models \phi \ .$$

Furthermore, since $U_\alpha = \bigcup_{\beta<\alpha} U_\beta$ and $\mathrm{pr}(\phi)$ is a finite set of predicate symbols, there is some $\beta < \alpha$ such that $\mathrm{pr}(\phi)$ is a subset of U_β. Consequently, by Observation 93(3), we obtain

$$\left(\mathcal{Y}^{[U_\beta]}, \sigma(\mathcal{M}, U_\alpha)^{[U_\beta]}\right) \not\models \phi \ .$$

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 128, $\mathcal{Y}_1$ is saturated relative to U_β and $\mathcal{Y}_2$ is saturated relative to $U_\alpha \setminus U_\beta$ and thus by Observation 95(6) also relative to $\mathcal{P} \setminus U_\beta$. Furthermore, $\mathcal{Y} = \mathcal{Y}_1 \cap \mathcal{Y}_2$, so, by Observation 97(2), $\mathcal{Y}^{[U_\beta]} = \mathcal{Y}_1^{[U_\beta]}$. 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

$$\left(\sigma(\mathcal{M}, U_\beta)^{[U_\beta]}, \sigma(\mathcal{M}, U_\alpha)^{[U_\beta]}\right) \not\models \phi \ .$$

Finally, since U_β is a subset of U_α, Observation 95(5) implies that

$$\sigma(\mathcal{M}, U_\alpha)^{[U_\beta]} = \mathcal{M}^{[U_\beta]} = \sigma(\mathcal{M}, U_\beta)^{[U_\beta]} \ .$$

Therefore,

$$\left(\sigma(\mathcal{M}, U_\beta)^{[U_\beta]}, \sigma(\mathcal{M}, U_\beta)^{[U_\beta]}\right) \not\models \phi \ .$$

Observation 93(3) 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 124, $\sigma(\mathcal{M}, U_\beta)$ 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 130. *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 124 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

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \ .$$

By Corollary 124, $\mathcal{N} = \sigma(\mathcal{M}, U_{\alpha+1})$ 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 124, 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 Observation 95(4),

$$\sigma(\mathcal{N}, U_\alpha) = \sigma(\sigma(\mathcal{M}, U_{\alpha+1}), U_\alpha) = \sigma(\mathcal{M}, U_{\alpha+1} \cap U_\alpha) = \sigma(\mathcal{M}, U_\alpha)$$

and also

$$\sigma(\mathcal{N}, \mathcal{P} \setminus U_\alpha) = \sigma(\sigma(\mathcal{M}, U_{\alpha+1}), \mathcal{P} \setminus U_\alpha) = \sigma(\mathcal{M}, U_{\alpha+1} \cap (\mathcal{P} \setminus U_\alpha))$$
$$= \sigma(\mathcal{M}, U_{\alpha+1} \setminus U_\alpha) \enspace .$$

Lemma 129 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

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \bigcap_{\beta \leq \alpha} \mathcal{X}_\beta\right) \enspace .$$

Third, for every limit ordinal $\alpha < \mu$, $\mathcal{X}_\alpha = \mathcal{I}_\mathcal{L}$ 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 Observation 95(3) that $\mathcal{M}$ is a subset of $\mathcal{X}_\alpha$ for every $\alpha < \mu$. Hence,

$$\emptyset \neq \mathcal{M} \subseteq \bigcap_{\alpha<\mu} \mathcal{X}_\alpha \enspace . \qquad \square$$

Proposition 131. *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

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \mathcal{Y}_\alpha\right) \enspace .$$

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 123, $\mathcal{Y}_\alpha \cap \mathcal{X}_{\alpha+1} = \mathcal{Y}_{\alpha+1}$ 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 $\mathrm{pr}(\phi) \subseteq U_\beta$. 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 Observation 95(2), $\mathcal{X}_{\gamma+1}$ is saturated relative to $U_{\gamma+1} \setminus U_\gamma$. Consequently, by Lemma 128, $\mathcal{Y}_\beta = \bigcap_{\gamma \leq \beta} \mathcal{X}_\gamma$ 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 Observation 95(6), it is also saturated relative to $\mathcal{P} \setminus U_\beta$. Hence, by Observation 97(2), 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 Observation 93(2).

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

$$\mathcal{Y}''^{[U_\beta]} = \mathcal{Y}'^{[U_\beta]} \cup \mathcal{Y}_\beta^{[U_\beta]} = \mathcal{Y}'^{[U_\beta]} \cup \mathcal{Y}_\alpha^{[U_\beta]} = (\mathcal{Y}' \cup \mathcal{Y}_\alpha)^{[U_\beta]} = \mathcal{Y}'^{[U_\beta]} \enspace .$$

Consequently, by Observation 93(3), $(\mathcal{Y}', \mathcal{Y}_\alpha) \not\models \phi$. $\square$

Lemma 132. *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 $\mathcal{Y}_\mu = \mathcal{I}_\mathcal{L}$, 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 $\mathcal{Y}_\mu = \mathcal{I}_\mathcal{L}$, 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

$$e_{V_\beta}\left(b_{V_\mu}(\mathcal{K}), \bigcap_{\gamma\leq\beta} \mathcal{Y}_\gamma\right) = e_\mathcal{P}\left(\mathcal{K}, \bigcap_{\gamma\leq\beta} \mathcal{Y}_\gamma\right) = \emptyset \enspace .$$

Consequently, $\mathcal{Y}_\mu = \mathcal{I}_\mathcal{L}$ is its MKNF model. $\square$

Theorem 133 *(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 130 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 $\mathcal{Y}_\mu = \mathcal{I}_\mathcal{L}$. By Lemma 132, $\boldsymbol{V}$ is a splitting sequence for $\mathcal{K}$ and $\boldsymbol{Y}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{V}$. Thus, by Lemma 129,

$$\bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \bigcap_{\alpha<\mu} \mathcal{Y}_\alpha = \mathcal{I}_\mathcal{L} \cap \bigcap_{\alpha<\mu} \mathcal{Y}_\alpha = \bigcap_{\alpha<\mu+1} \mathcal{Y}_\alpha = \sigma(\mathcal{M}, V_\mu) = \sigma(\mathcal{M}, \mathcal{P}) = \mathcal{M} \ .$$

To prove the converse implication, suppose that $\boldsymbol{X}$ is a solution to $\mathcal{K}$ w.r.t. $\boldsymbol{U}$. Then, by Lemma 132, 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,

$$\bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \bigcap_{\alpha\le\mu} \mathcal{Y}_\alpha$$

and, by Proposition 131, $\bigcap_{\alpha\le\mu} \mathcal{Y}_\alpha$ is an MKNF model of $b_{V_\mu}(\mathcal{K}) = b_\mathcal{P}(\mathcal{K}) = \mathcal{K}$. □

Theorem 54 *(Splitting theorem for MKNF knowledge bases). The MKNF models semantics for MKNF knowledge bases satisfies the Abstract Splitting Set Property and the Abstract Splitting Sequence Property.*

Proof. Follows from Theorems 123 and 133. □

C.2. Ontology updates

Proposition 134. *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,*

$$[\![\mathcal{T}]\!] = \bigcap_{\alpha<\mu} [\![b_{A_\alpha}(\mathcal{T})]\!] \ .$$

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 135. *Let $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. Then the following holds:*

(1) *If $\mathcal{M} \subseteq \mathcal{N}$, then $\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N} = \mathcal{M}$.*
(2) *If $\mathcal{M} \supseteq \mathcal{N}$, then $\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N} = \mathcal{N}$.*
(3) *$\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{M} = \mathcal{M}$.*
(4) *$\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N} = \emptyset$ if and only if $\mathcal{M} = \emptyset$ or $\mathcal{N} = \emptyset$.*

Proof.

(1) For every interpretation I, $I <^I_\mathsf{W} J$ for every $J \neq I$. Thus, if $\mathcal{M} \subseteq \mathcal{N}$, for all $I \in \mathcal{M}$ it holds that $I \blacklozenge_\mathsf{W} \mathcal{N} = \{ I \}$. Therefore, $\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N} = \mathcal{M}$.
(2) For every interpretation I, $I <^I_\mathsf{W} J$ 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 $\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N}$. Furthermore, $\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N} \subseteq \mathcal{N}$ by construction, so we obtain $\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N} = \mathcal{N}$.
(3) Follows from (1).
(4) The direct implication follows by definition of $\blacklozenge_\mathsf{W}$. The converse implication follows from (1) and (2). □

Proposition 136. *Let $\boldsymbol{T} = \langle \mathcal{T}_i \rangle_{i<n}$ be a sequence of first-order theories with $n > 0$ and $\mathcal{M} = [\![\Diamond_\mathsf{W} \boldsymbol{T}]\!]$. Then $\mathcal{M} \models \mathcal{T}_{n-1}$.*

Proof. Follows by induction on n, due to the fact that $\mathcal{M} \blacklozenge_\mathsf{W} \mathcal{N} \subseteq \mathcal{N}$ for any $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$. □

Proposition 137. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha<\mu}$ be a saturation sequence, $I, J \in \mathcal{I}_\mathcal{L}$ and $\mathcal{N} \in \mathfrak{M}$ be sequence-saturated relative to $\boldsymbol{A}$. Then,*

$$J \in (I \blacklozenge_\mathsf{W} \mathcal{N}) \qquad \text{if and only if} \qquad \forall \alpha < \mu : J^{[A_\alpha]} \in \left(I^{[A_\alpha]} \blacklozenge_\mathsf{W} \mathcal{N}^{[A_\alpha]} \right) \ .$$

Proof. Suppose that $J \notin (I \blacklozenge_{\mathrm{W}} \mathcal{N})$. 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 $J^{[A_\alpha]} \notin \left(I^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right)$, 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

$$\left(P^{K^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) = \left(P^K \div P^I\right) \subseteq \left(P^J \div P^I\right) = \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) \ .$$

Finally, for any predicate symbol P that does not belong to A_α,

$$P^{I^{[A_\alpha]}} = P^{J^{[A_\alpha]}} = P^{K^{[A_\alpha]}} = \emptyset \ .$$

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

$$K^{[A_\alpha]} <^{I^{[A_\alpha]}}_{\mathrm{W}} J^{[A_\alpha]} \ ,$$

so $J^{[A_\alpha]} \notin \left(I^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right)$ as desired.

For the converse implication, suppose that for some $\alpha < \mu$, $J^{[A_\alpha]} \notin \left(I^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right)$. If $J^{[A_\alpha]} \notin \mathcal{N}^{[A_\alpha]}$, we immediately obtain that $J \notin \mathcal{N}$. Consequently, $J \notin (I \blacklozenge_{\mathrm{W}} \mathcal{N})$.

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 $K <^{I^{[A_\alpha]}}_{\mathrm{W}} J^{[A_\alpha]}$. Thus, for all predicate symbols $P \in \mathcal{P}$ we know that

$$\left(P^K \div P^{I^{[A_\alpha]}}\right) \subseteq \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) \ . \tag{C.10}$$

We also know that there is some predicate symbol $P_0 \in \mathcal{P}$ such that

$$\left(P_0^K \div P_0^{I^{[A_\alpha]}}\right) \subsetneq \left(P_0^{J^{[A_\alpha]}} \div P_0^{I^{[A_\alpha]}}\right) \ . \tag{C.11}$$

Additionally, for every predicate symbol P from $\mathcal{P} \setminus A_\alpha$ it holds that

$$P^{I^{[A_\alpha]}} = P^{J^{[A_\alpha]}} = P^K = \emptyset \ .$$

Thus,

$$\left(P^K \div P^{I^{[A_\alpha]}}\right) = \emptyset = \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) \ .$$

Consequently, $P_0 \in A_\alpha$. Let K' be an interpretation such that for every ground atom p,

$$K' \models p \qquad \text{if and only if} \qquad K \models p \vee J^{[\mathcal{P} \setminus A_\alpha]} \models p \ .$$

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

$$\left(P^{K'} \div P^I\right) = \left(P^K \div P^{I^{[A_\alpha]}}\right) \subseteq \left(P^{J^{[A_\alpha]}} \div P^{I^{[A_\alpha]}}\right) = \left(P^J \div P^I\right) \ .$$

b) If $P \in \mathcal{P} \setminus A_\alpha$, then since $K'^{[\mathcal{P} \setminus A_\alpha]} = J^{[\mathcal{P} \setminus A_\alpha]}$,

$$\left(P^{K'} \div P^I\right) = \left(P^J \div P^I\right) .$$

Moreover, from (C.11) we obtain

$$\left(P_0^{K'} \div P_0^I\right) = \left(P_0^K \div P_0^{I^{[A_\alpha]}}\right) \subsetneq \left(P_0^{J^{[A_\alpha]}} \div P_0^{I^{[A_\alpha]}}\right) = \left(P_0^J \div P_0^I\right) .$$

It follows from the above considerations that $K' <_{\mathrm{W}}^I J$. Consequently, $J \notin (I \blacklozenge_{\mathrm{W}} \mathcal{N})$. □

Proposition 138. *Let $\boldsymbol{A} = \langle A_\alpha \rangle_{\alpha < \mu}$ be a saturation sequence, $J \in \mathfrak{I}_{\mathcal{L}}$ and $\mathcal{M}, \mathcal{N} \in \mathfrak{M}$ be both sequence-saturated relative to $\boldsymbol{A}$. Then,*

$$J \in (\mathcal{M} \blacklozenge_{\mathrm{W}} \mathcal{N}) \qquad \textit{if and only if} \qquad \forall \alpha < \mu : J^{[A_\alpha]} \in \left(\mathcal{M}^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right) .$$

Proof. By definition, $J \in (\mathcal{M} \blacklozenge_{\mathrm{W}} \mathcal{N})$ if and only if for some $I \in \mathcal{M}$, $J \in (I \blacklozenge_{\mathrm{W}} \mathcal{N})$. By Proposition 137, this holds if and only if

$$\forall \alpha < \mu : J^{[A_\alpha]} \in \left(I^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right) . \tag{C.12}$$

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:

$$\forall \alpha < \mu : J^{[A_\alpha]} \in \left(I_\alpha^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right) . \tag{C.13}$$

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,

$$I \models p \qquad \text{if and only if} \qquad \exists \alpha < \mu : I_\alpha^{[A_\alpha]} \models p .$$

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, $J^{[A_\alpha]} \in \left(I^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right)$. As a consequence, (C.12) is satisfied and our proof is finished. □

Proposition 139. *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,*

$$J^{[A_\alpha]} \in \left(\mathcal{M}^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right) \qquad \textit{if and only if} \qquad J \in (\sigma(\mathcal{M}, A_\alpha) \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)) .$$

Proof. First note that if $\mathcal{M} = \emptyset$ or $\mathcal{N} = \emptyset$, then the equivalence is trivially satisfied since both $(\mathcal{M}^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}) = \emptyset$ and $(\sigma(\mathcal{M}, A_\alpha) \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)) = \emptyset$. Thus, we can assume that $\mathcal{M}$ and $\mathcal{N}$ are MKNF interpretations.

By applying Proposition 138 to $\sigma(\mathcal{M}, A_\alpha)$ and $\sigma(\mathcal{N}, A_\alpha)$ it follows that for every interpretation J,

$$J \in (\sigma(\mathcal{M}, A_\alpha) \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)) \qquad \text{if and only if} \qquad \forall \beta < \mu : J^{[A_\beta]} \in \left(\sigma(\mathcal{M}, A_\alpha)^{[A_\beta]} \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)^{[A_\beta]}\right) . \tag{C.14}$$

By Observation 95(7) we obtain that whenever $\beta \neq \alpha$, $\sigma(\mathcal{M}, A_\alpha)^{[A_\beta]} = \mathfrak{I}_{\mathcal{L}}^{[A_\beta]}$ and $\sigma(\mathcal{N}, A_\alpha)^{[A_\beta]} = \mathfrak{I}_{\mathcal{L}}^{[A_\beta]}$, so by Lemma 135(3) we can conclude that $\sigma(\mathcal{M}, A_\alpha)^{[A_\beta]} \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)^{[A_\beta]} = \mathfrak{I}_{\mathcal{L}}^{[A_\beta]}$. Thus, condition (C.14) gets simplified to

$$J \in (\sigma(\mathcal{M}, A_\alpha) \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)) \qquad \text{if and only if} \qquad J^{[A_\alpha]} \in \left(\sigma(\mathcal{M}, A_\alpha)^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)^{[A_\alpha]}\right) .$$

Furthermore, by Observation 95(5), $\sigma(\mathcal{M}, A_\alpha)^{[A_\alpha]} = \mathcal{M}^{[A_\alpha]}$ and $\sigma(\mathcal{N}, A_\alpha)^{[A_\alpha]} = \mathcal{N}^{[A_\alpha]}$, so

$$J \in (\sigma(\mathcal{M}, A_\alpha) \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha)) \qquad \text{if and only if} \qquad J^{[A_\alpha]} \in \left(\mathcal{M}^{[A_\alpha]} \blacklozenge_{\mathrm{W}} \mathcal{N}^{[A_\alpha]}\right) .$$

This completes our proof. □

Corollary 140. *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,*

$$\mathcal{M} \blacklozenge_{\mathrm{W}} \mathcal{N} = \bigcap_{\alpha < \mu} \sigma(\mathcal{M}, A_\alpha) \blacklozenge_{\mathrm{W}} \sigma(\mathcal{N}, A_\alpha) .$$

Proof. Follows from Propositions 138 and 139. □

Lemma 141. *Let A be a set of predicate symbols, $I, J \in \mathcal{I}_{\mathcal{L}}$ and $\mathcal{N} \in \mathcal{M}$ be saturated relative to A. If $J \in I \blacklozenge_W \mathcal{N}$, then I coincides with J on $\mathcal{P} \setminus A$.*

Proof. If $J \in I \blacklozenge_W \mathcal{N}$, then $J \in \mathcal{N}$. Let J' be an interpretation such that for every ground atom p,

$$J' \models p \qquad \text{if and only if} \qquad J^{[A]} \models p \vee I^{[\mathcal{P}\setminus A]} \models p \ .$$

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 $J' <^I_W J$ holds, contrary to the assumption that J belongs to $I \blacklozenge_W \mathcal{N}$. 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 142. *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,*

$$[\![\Diamond_W \boldsymbol{T}]\!] = \bigcap_{\alpha<\mu} [\![\Diamond_W b_{A_\alpha}(\boldsymbol{T})]\!] \ .$$

Proof. We prove by induction on n.

1° If $n = 1$, then $[\![\Diamond_W \boldsymbol{T}]\!] = [\![\mathcal{T}_0]\!]$ and for every $\alpha < \mu$, $[\![\Diamond_W b_{A_\alpha}(\boldsymbol{T})]\!] = [\![b_{A_\alpha}(\mathcal{T}_0)]\!]$. The claim thus follows from Proposition 134.

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 $\Diamond_W$ we obtain

$$\left(\Diamond_W \boldsymbol{T}'\right) = \left(\Diamond_W \boldsymbol{T}\right) \diamond_W \mathcal{T}_n \qquad \text{and} \qquad \left(\Diamond_W b_{A_\alpha}(\boldsymbol{T}')\right) = \left(\Diamond_W b_{A_\alpha}(\boldsymbol{T})\right) \diamond_W b_{A_\alpha}(\mathcal{T}_n) \ .$$

Let $\mathcal{M} = [\![\Diamond_W \boldsymbol{T}]\!]$, $\mathcal{N} = [\![\mathcal{T}_n]\!]$ and for every $\alpha < \mu$, $\mathcal{M}_\alpha = [\![\Diamond_W b_{A_\alpha}(\boldsymbol{T})]\!]$ and $\mathcal{N}_\alpha = [\![b_{A_\alpha}(\mathcal{T}_n)]\!]$. Our goal is to prove that

$$\mathcal{M} \blacklozenge_W \mathcal{N} = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha \blacklozenge_W \mathcal{N}_\alpha \ .$$

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 134, respectively. Also, due to Theorem 71 and Proposition 69, 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 135(4), $\mathcal{M} \blacklozenge_W \mathcal{N} = \emptyset$ and it follows from Observation 100(2) that for some $\alpha < \mu$, either $\mathcal{M}_\alpha = \emptyset$ or $\mathcal{N}_\alpha = \emptyset$. In either case $\mathcal{M}_\alpha \blacklozenge_W \mathcal{N}_\alpha = \emptyset$ and the desired equation is satisfied.
In the principal case, both $\mathcal{M} \neq \emptyset$ and $\mathcal{N} \neq \emptyset$. Thus, we can use Observation 100(2) 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 Observation 100(1), $\mathcal{M}$ and $\mathcal{N}$ are sequence-saturated relative to $\boldsymbol{A}$. Furthermore, we can apply Corollary 140 to obtain the desired equation:

$$\mathcal{M} \blacklozenge_W \mathcal{N} = \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha) \blacklozenge_W \sigma(\mathcal{N}, A_\alpha) = \bigcap_{\alpha<\mu} \mathcal{M}_\alpha \blacklozenge_W \mathcal{N}_\alpha \ . \qquad \square$$

Theorem 57 *(Splitting theorem for Winslett's first-order operator). The semantics for sequences of first-order theories induced by Winslett's first-order operator $\diamond_W$ satisfies the Abstract Splitting Set Property and the Abstract Splitting Sequence Property.*

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 $\mathcal{X} \in \mathcal{M}$ and all ordinals α with $\alpha + 1 < \mu$:

$$b_{A_0}(\boldsymbol{T}) = b_{U_0}(\boldsymbol{T}) \ ,$$
$$b_{A_{\alpha+1}}(\boldsymbol{T}) = b_{U_{\alpha+1} \setminus U_\alpha}(\boldsymbol{T}) = t_{U_\alpha}(b_{U_\alpha+1}(\boldsymbol{T})) = e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{T}), \mathcal{X}) \ .$$

Therefore, the Abstract Splitting Sequence Property follows from Proposition 142. The Abstract Splitting Set Property for a splitting set U follows from the splitting sequence property applied to the splitting sequence $\langle U, \mathcal{P} \rangle$. □

C.3. Rule updates

Theorem 60 *(Splitting theorem for rule updates). The* RD*-semantics for rule updates satisfies the Abstract Splitting Set Property and the Abstract Splitting Sequence Property.*

Proof (sketch). Let $\boldsymbol{P}$ be a DLP. We need to prove that J is an RD-model of $\boldsymbol{P}$ if and only if $J = \bigcup_{\alpha<\mu} J_\alpha$ where for every $\alpha < \mu$, J_α is an RD-model of $\boldsymbol{P}_\alpha$ and

- $\boldsymbol{P}_0 = b_{U_0}(\boldsymbol{P})$.
- For any ordinal α such that $\alpha + 1 < \mu$, $\boldsymbol{P}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{P}), \bigcup_{\beta \le \alpha} J_\beta\right)$.
- For any limit ordinal $\alpha < \mu$, $\boldsymbol{P}_\alpha = \langle \emptyset \rangle_{\alpha < \mu}$.

This essentially follows from the splitting properties of logic programs [71] and from the observation that rules in $\mathsf{rej}(\boldsymbol{P}, J)$ correspond one-to-one with the rules in $\bigcup_{\alpha<\mu} \mathsf{rej}(\boldsymbol{P}_\alpha, J_\alpha)$. In particular, if we put

$$\mathcal{Q} = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)] \qquad \text{and} \qquad \mathcal{Q}_\alpha = [\mathsf{all}(\boldsymbol{P}_\alpha) \setminus \mathsf{rej}(\boldsymbol{P}_\alpha, J_\alpha)] \enspace ,$$

then it follows that

- $\mathcal{Q}_0 = b_{U_0}(\mathcal{Q})$.
- For every ordinal α such that $\alpha + 1 < \mu$, $\mathcal{Q}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{Q}), \bigcup_{\beta \le \alpha} J_\beta\right)$.
- For every limit ordinal α, $\mathcal{Q}_\alpha = \emptyset$.

What remains is to additionally handle the set of default assumptions. 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

$$\mathsf{def}(\boldsymbol{Q}, K, B) = \{\, \sim l \mid l \in \mathit{Lits}_\mathsf{G} \wedge \mathsf{pr}(l) \subseteq B \wedge \neg(\exists \pi \in \mathsf{all}(\boldsymbol{Q}) : \mathsf{H}_\pi = l \wedge K \models \mathsf{B}_\pi) \,\} \enspace .$$

It is not difficult to se 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 RD-model of $\boldsymbol{Q}$, if accompanied by a suitable restriction in the definition of K', i.e. $K' = K \cup \{\, \sim l \mid l \in \mathit{Lits}_\mathsf{G} \setminus K \wedge \mathsf{pr}(l) \subseteq B \,\}$. Let

$$\mathcal{Q}' = [\mathsf{all}(\boldsymbol{P}) \setminus \mathsf{rej}(\boldsymbol{P}, J)] \cup \mathsf{def}(\boldsymbol{P}, J, \mathcal{P}) \enspace ,$$
$$\mathcal{Q}'_\alpha = [\mathsf{all}(\boldsymbol{P}_\alpha) \setminus \mathsf{rej}(\boldsymbol{P}_\alpha, J_\alpha)] \cup \mathsf{def}(\boldsymbol{P}_\alpha, J_\alpha, A_\alpha) \enspace .$$

Similarly as before, it follows that

- $\mathcal{Q}'_0 = b_{U_0}(\mathcal{Q}')$.
- For every ordinal α such that $\alpha + 1 < \mu$, $\mathcal{Q}'_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{Q}'), \bigcup_{\beta \le \alpha} J_\beta\right)$.
- For every limit ordinal α, $\mathcal{Q}'_\alpha = \emptyset$.

Hence, by the results of [71], $J' = \mathsf{least}(\mathcal{Q}')$ 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 RD-models. □

Appendix D. Proofs: layered dynamic MKNF knowledge bases

In the following we present proofs of results from Section 5, implicitly working under the same assumptions as those imposed in that section. 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. In the proofs we also implicitly use many of the results presented in Appendix A.

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. For the sake of completeness, we formulate it here once again, 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*. We note that the use of limit ordinals is only required to fit the abstract definition (Definition 49). Note however, that because layerings of DMKBs are based on predicate symbols, any finite DMKB will have a finite layering splitting sequence.

Definition 143 *(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
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right) \enspace .$$
3. For any limit ordinal α, $\mathcal{X}_\alpha = \mathfrak{I}_\mathcal{L}$.
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. $\boldsymbol{U}$.

D.1. Independence of splitting sequence

Theorem 66 *(Language conservation and fact update for rule updates). The* RD*-semantics for rule updates conserves the language and respects fact update.*

Proof (sketch). Language conservation is an immediate consequence of the definition of the RD-semantics and of basic properties of the least model of a definite program.

As for fact update, let $\boldsymbol{P} = \langle \mathcal{P}_i \rangle_{i<n}$ be a sequence of consistent sets of facts. 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:

$$\{\, l. \mid \exists j < n : (l.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow (\sim l.) \notin \mathcal{P}_i^{\mathsf{e}}) \,\} \cup \left\{ \sim l. \mid \exists j < n : (\sim l.) \in \mathcal{P}_j^{\mathsf{e}} \land (\forall i : j < i < n \Longrightarrow (l.) \notin \mathcal{P}_i) \right\} \tag{D.1}$$

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 RD-model

$$\left\{ l \mid \exists j < n : (l.) \in \mathcal{P}_j \land (\forall i : j < i < n \Longrightarrow \left\{ \bar{l}., \sim l. \right\} \cap \mathcal{P}_i = \emptyset) \right\} . \qquad \square$$

Proposition 69. *Let A be a set of predicate symbols and $\mathcal{T}$ a first-order theory such that* $\mathrm{pr}(\mathcal{T}) \subseteq A$*. 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 Observation 95(2). $\square$

Definition 144 *(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:

$$I \blacklozenge_{\mathsf{W}} \mathcal{N} = \min\left(\mathcal{N}, \leq^I_{\mathsf{W}}\right) ,$$

$$\mathcal{M} \blacklozenge_{\mathsf{W}} \mathcal{N} = \bigcup_{I \in \mathcal{M}} I \blacklozenge_{\mathsf{W}} \mathcal{N} = \bigcup_{I \in \mathcal{M}} \min\left(\mathcal{N}, \leq^I_{\mathsf{W}}\right) .$$

Remark 145. Note that the above definition is compatible with the definition of Winslett's operator $\diamond_{\mathsf{W}}$ defined in Section 2.5, 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 146. *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,

$$I' \models p \qquad \text{if and only if} \qquad I^{[A]} \models p \lor J^{[\mathcal{P} \setminus A]} \models p .$$

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 $J'' <^{I'}_{\mathsf{W}} J$. This means that for every predicate symbol $P \in A$,

$$P^{J''} \div P^{I'} \subseteq P^J \div P^{I'} \tag{D.2}$$

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,

$$P_0^{J''} \div P_0^{I'} \subsetneq P_0^J \div P_0^{I'} .$$

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 (D.2), 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, $J''' <_{\mathsf{W}}^{I} J'$. 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 71 *(Language conservation and fact update for ontology updates). Winslett's first-order update operator $\diamond_{\mathsf{W}}$ conserves the language and respects fact update.*

Proof. Language conservation follows by induction on i from Proposition 69 and Lemma 146.

As for fact update, 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 $[\![\diamond_{\mathsf{W}} \boldsymbol{T}]\!] = [\![\mathcal{T}_0]\!] = [\![\{ l \in \mathit{Lits}_{\mathsf{G}} \mid l \in \mathcal{T}_0 \}]\!]$, 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 $\diamond_{\mathsf{W}}$,

$$\mathcal{M}' = [\![\diamond_{\mathsf{W}} \boldsymbol{T}']\!] = \bigcup_{I \in \mathcal{M}} \min\left([\![\mathcal{T}_n]\!], \leq_{\mathsf{W}}^{I}\right) \enspace . \tag{D.3}$$

By the inductive assumption for $\boldsymbol{T}$ we obtain that $I \in \mathcal{M}$ if and only if

$$I \models \left\{ l \in \mathit{Lits}_{\mathsf{G}} \;\middle|\; \exists j < n : l \in \mathcal{T}_j \land (\forall i : j < i < n \Longrightarrow \bar{l} \notin \mathcal{T}_i) \right\} \enspace . \tag{D.4}$$

Our goal is to prove that $I' \in \mathcal{M}'$ if and only if

$$I' \models \left\{ l \in \mathit{Lits}_{\mathsf{G}} \;\middle|\; \exists j < n+1 : l \in \mathcal{T}_j \land (\forall i : j < i < n+1 \Longrightarrow \bar{l} \notin \mathcal{T}_i) \right\} \enspace . \tag{D.5}$$

Take some $I' \in \mathcal{M}'$. Then it follows from (D.3) that there is some $I \in \mathcal{M}$ such that $I' \in \min\left([\![\mathcal{T}_n]\!], \leq_{\mathsf{W}}^{I}\right)$. Hence, $I' \models \{ l \in \mathit{Lits}_{\mathsf{G}} \mid l \in \mathcal{T}_n \}$ 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 (D.4), we conclude that I' satisfies (D.5).
For the converse inclusion, suppose that I' satisfies (D.5) and let I be an interpretation that satisfies (D.4) and interprets a minimal set of ground atoms differently from I'. By (D.5) 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 (D.4), it follows that $I \in \mathcal{M}$. This implies that $I' \in \mathcal{M}'$. □

D.2. Splitting-based updates of MKNF knowledge bases

Proposition 147. *Let $\boldsymbol{K}$ be a basic DMKB and $\boldsymbol{U}$ a splitting sequence for $\boldsymbol{K}$ and suppose that both $\diamond$ and S have Abstract 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 Abstract 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 Abstract 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

$$\mathcal{M} = \{ I \in \mathcal{I}_{\mathcal{L}} \mid I \models J \} \qquad \text{and} \qquad \forall \alpha < \mu : \mathcal{X}_\alpha = \{ I \in \mathcal{I}_{\mathcal{L}} \mid I \models J_\alpha \} \enspace .$$

Consequently,

$$\mathcal{M} = \{ I \in \mathcal{I}_{\mathcal{L}} \mid I \models J \} = \left\{ I \in \mathcal{I}_{\mathcal{L}} \;\middle|\; I \models \textstyle\bigcup_{\alpha<\mu} J_\alpha \right\} = \bigcap_{\alpha<\mu} \{ I \in \mathcal{I}_{\mathcal{L}} \mid I \models J_\alpha \} = \bigcap_{\alpha<\mu} \mathcal{X}_\alpha \enspace . \qquad \square$$

Proposition 148. *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

$$\mathcal{M} = \{ \, I \in \mathcal{I}_{\mathcal{L}} \mid I \models J \, \}$$

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 $\mathcal{M}$, $I \in \mathcal{M}$. □

Corollary 149. *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 Abstract 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 147, 148, and Observation 100(1). □

Lemma 150. *Let $\boldsymbol{K}$ be a DMKB and U, V some sets of predicate symbols. Then,*

$$b_U(b_V(\boldsymbol{K})) = b_{U \cap V}(\boldsymbol{K}) \ .$$

Proof. Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i) \rangle_{i<n}$. We obtain the following:

$$b_U(b_V(\boldsymbol{K})) = \langle (b_U(b_V(\mathcal{O}_i)), b_U(b_V(\mathcal{P}_i))) \rangle_{i<n}$$
$$b_{U \cap V}(\boldsymbol{K}) = \langle (b_{U \cap V}(\mathcal{O}_i), b_{U \cap V}(\mathcal{P}_i)) \rangle_{i<n}$$

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,

$$\begin{aligned} \phi \in b_U(b_V(\mathcal{O}_i)) &\iff \phi \in b_V(\mathcal{O}_i) \land \mathsf{pr}(\phi) \subseteq U \iff \phi \in \mathcal{O}_i \land \mathsf{pr}(\phi) \subseteq V \land \mathsf{pr}(\phi) \subseteq U \\ &\iff \phi \in \mathcal{O}_i \land \mathsf{pr}(\phi) \subseteq U \cap V \iff \phi \in b_{U \cap V}(\mathcal{O}_i) \ . \end{aligned}$$

Similarly, by the definition of the bottom of a program,

$$\begin{aligned} \pi \in b_U(b_V(\mathcal{P}_i)) &\iff \pi \in b_V(\mathcal{P}_i) \land \mathsf{pr}(\pi) \subseteq U \iff \pi \in \mathcal{P}_i \land \mathsf{pr}(\pi) \subseteq V \land \mathsf{pr}(\pi) \subseteq U \\ &\iff \pi \in \mathcal{P}_i \land \mathsf{pr}(\pi) \subseteq U \cap V \iff \pi \in b_{U \cap V}(\mathcal{P}_i) \ . \qquad \square \end{aligned}$$

Lemma 151. *Let $\boldsymbol{K}$ be a DMKB, $\mathcal{X} \in \mathfrak{M}$, U a set of predicate symbols and V a splitting set for $\boldsymbol{K}$. Then,*

$$e_U(b_V(\boldsymbol{K}), \mathcal{X}) = b_V(e_U(\boldsymbol{K}, \mathcal{X})) \ .$$

Proof. Let $\boldsymbol{K} = \langle (\mathcal{O}_i, \mathcal{P}_i)_{i<n} \rangle$. We obtain:

$$e_U(b_V(\boldsymbol{K}), \mathcal{X}) = \langle (t_U(b_V(\mathcal{O}_i)), e_U(b_V(\mathcal{P}_i), \mathcal{X})) \rangle_{i<n}$$
$$b_V(e_U(\boldsymbol{K}, \mathcal{X})) = \langle (b_V(t_U(\mathcal{O}_i)), b_V(e_U(\mathcal{P}_i, \mathcal{X}))) \rangle_{i<n}$$

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,

$$\begin{aligned} \phi \in t_U(b_V(\mathcal{O}_i)) &\iff \phi \in b_V(\mathcal{O}_i) \land \mathsf{pr}(\phi) \nsubseteq U \iff \phi \in \mathcal{O}_i \land \mathsf{pr}(\phi) \subseteq V \land \mathsf{pr}(\phi) \nsubseteq U \\ &\iff \phi \in t_U(\mathcal{O}_i) \land \mathsf{pr}(\phi) \subseteq V \iff \phi \in b_V(t_U(\mathcal{O}_i)) \ . \end{aligned}$$

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

$$\begin{aligned} \mathsf{H}_\sigma &= \mathsf{H}_\pi \ , & \mathcal{X} &\models \kappa(\{ \, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq U \, \}) \ , \\ \mathsf{B}_\sigma &= \{ \, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus U \, \} \ , & \mathsf{pr}(\pi) &\nsubseteq U \ . \end{aligned}$$

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

$$\begin{aligned} \mathsf{H}_\sigma &= \mathsf{H}_\pi \enspace, & \mathcal{X} &\models \kappa(\{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq U \,\}) \enspace, \\ \mathsf{B}_\sigma &= \{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus U \,\} \enspace, & \mathsf{pr}(\pi) &\nsubseteq U \enspace. \end{aligned}$$

Since $\mathsf{pr}(\mathsf{H}_\pi) \subseteq V$ 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 152. *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,

$$I' \models p \qquad \text{if and only if} \qquad I \models p \lor J \models p \enspace.$$

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 153. *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:

$$\begin{aligned} e_U(e_V(\boldsymbol{K}, \mathcal{Y}), \mathcal{X}) &= \langle (t_U(t_V(\mathcal{O}_i)), e_U(e_V(\mathcal{P}_i, \mathcal{Y}), \mathcal{X})) \rangle_{i<n} \\ e_{U \cup V}(\boldsymbol{K}, \mathcal{X} \cap \mathcal{Y}) &= \langle (t_{U \cup V}(\mathcal{O}_i), e_{U \cup V}(\mathcal{P}_i, \mathcal{X} \cap \mathcal{Y})) \rangle_{i<n} \end{aligned}$$

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 150,

$$t_U(t_V(\mathcal{O}_i)) = b_{\mathcal{P} \setminus U}(b_{\mathcal{P} \setminus V}(\mathcal{O}_i)) = b_{(\mathcal{P} \setminus U) \cap (\mathcal{P} \setminus V)}(\mathcal{O}_i) = b_{\mathcal{P} \setminus (U \cup V)}(\mathcal{O}_i) = t_{U \cup V}(\mathcal{O}_i) \enspace.$$

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 $\sigma' \in e_V(\mathcal{P}_i, \mathcal{Y})$,

$$\begin{aligned} [b]\mathsf{H}_\sigma &= \mathsf{H}_{\sigma'} \enspace, & \mathcal{X} &\models \kappa(\{\, L \in \mathsf{B}_{\sigma'} \mid \mathsf{pr}(L) \subseteq U \,\}) \enspace, \\ \mathsf{B}_\sigma &= \{\, L \in \mathsf{B}_{\sigma'} \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus U \,\} \enspace, & \mathsf{pr}(\sigma') &\nsubseteq U \enspace. \end{aligned} \tag{D.6}$$

Furthermore, $\sigma' \in e_V(\mathcal{P}_i, \mathcal{Y})$ if and only if for some rule $\pi \in \mathcal{P}_i$,

$$\begin{aligned} [b]\mathsf{H}_{\sigma'} &= \mathsf{H}_\pi \enspace, & \mathcal{Y} &\models \kappa(\{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq V \,\}) \enspace, \\ \mathsf{B}_{\sigma'} &= \{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus V \,\} \enspace, & \mathsf{pr}(\pi) &\nsubseteq V \enspace. \end{aligned} \tag{D.7}$$

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 $\mathsf{pr}(\mathsf{H}_{\sigma'}) \nsubseteq U$ and $\mathsf{pr}(\mathsf{H}_\pi) \nsubseteq V$, respectively. Also, since $\mathsf{pr}(\mathsf{H}_\pi)$ is a singleton set, together they are equivalent to $\mathsf{pr}(\mathsf{H}_\pi) \nsubseteq U \cup V$.

Moreover, by Lemma 152, $\mathcal{X} \cap \mathcal{Y}$ coincides with $\mathcal{X}$ on U and with $\mathcal{Y}$ on V. These observations imply that (D.6) and (D.7) are together equivalent to an existence of a rule $\pi \in \mathcal{P}_i$ such that

$$\begin{aligned} \mathsf{H}_\sigma &= \mathsf{H}_\pi \enspace, & \mathcal{X} \cap \mathcal{Y} &\models \kappa(\{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq U \cup V \,\}), \\ \mathsf{B}_\sigma &= \{\, L \in \mathsf{B}_\pi \mid \mathsf{pr}(L) \subseteq \mathcal{P} \setminus (U \cup V) \,\}, & \mathsf{pr}(\mathsf{H}_\pi) &\nsubseteq (U \cup V) \enspace. \end{aligned}$$

This is equivalent to $\sigma \in e_{U \cup V}(\mathcal{P}_i, \mathcal{X} \cap \mathcal{Y})$. □

Lemma 154. *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 Observation 95(5) 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 Observation 93(2),

$$\mathcal{X} \models \kappa(S) \Longleftrightarrow \sigma(\mathcal{X}, V) \models \kappa(S) \ . \qquad \square$$

Lemma 155. *Let $\boldsymbol{A}$ be a saturation sequence, $\mathcal{M} \in \mathfrak{M}$ 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,

$$K \models p \qquad \text{if and only if} \qquad \exists \alpha < \mu : K_\alpha^{[A_\alpha]} \models p \ .$$

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

$$K^{[A_\alpha \cap U]} = K_\alpha^{[A_\alpha \cap U]} = J^{[A_\alpha \cap U]} = I^{[A_\alpha \cap U]} \ .$$

Since every predicate symbol from U belongs to A_α for some $\alpha < \mu$, we can conclude that $K^{[U]} = I^{[U]}$. Consequently, $I \in \sigma(\mathcal{M}, U)$. $\square$

Lemma 156. *Let $\boldsymbol{U} = \langle U_\alpha \rangle_{\alpha < \mu}$ be a splitting sequence, $\boldsymbol{A}$ the saturation sequence induced by $\boldsymbol{U}$ and $\mathcal{M} \in \mathfrak{M}$ be sequence-saturated relative to $\boldsymbol{A}$. Then for any ordinal $\alpha < \mu$,*

$$\sigma\left(\mathcal{M}, \textstyle\bigcup_{\beta \leq \alpha} A_\beta\right) = \sigma(\mathcal{M}, U_\alpha) = \bigcap_{\beta \leq \alpha} \sigma(\mathcal{M}, A_\beta) \ .$$

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,

$$\bigcup_{\beta \leq \alpha+1} A_\beta = U_\alpha \cup A_{\alpha+1} = U_\alpha \cup (U_{\alpha+1} \setminus U_\alpha) = U_\alpha \cup U_{\alpha+1} = U_{\alpha+1} \ .$$

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,

$$\bigcup_{\beta \leq \alpha} A_\beta = A_\alpha \cup \bigcup_{\beta < \alpha} A_\beta = \emptyset \cup \bigcup_{\beta < \alpha} \bigcup_{\gamma \leq \beta} A_\gamma = \bigcup_{\beta < \alpha} U_\beta = U_\alpha \ .$$

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,

$$J \models p \qquad \text{if and only if} \qquad I^{[U_\alpha]} \models p \vee J_0^{[\mathcal{P} \setminus U_\alpha]} \models p \ .$$

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 $I \in \sigma(\mathcal{M}, U_\alpha)$. $\square$

Proposition 72 *(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 Abstract 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
$$\boldsymbol{K}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\gamma \le \alpha} \mathcal{X}_\gamma\right) \enspace .$$
- For any limit ordinal $\alpha < \mu$, $\mathcal{X}_\alpha = \mathcal{I}_\mathcal{L}$, 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 148, $\mathcal{X}_\alpha$ is saturated relative to A_α. Thus, by Observation 100(2), $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$. Moreover, by Lemma 156, $\bigcap_{\gamma \le \alpha} \mathcal{X}_\gamma = \bigcap_{\gamma \le \alpha} \sigma(\mathcal{M}, A_\gamma) = \sigma(\mathcal{M}, U_\alpha)$, so
$$\boldsymbol{K}_{\alpha+1} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha)\right) \enspace .$$

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 147 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
$$\boldsymbol{K}_{\alpha,\beta+1} = e_{V_\beta}\left(b_{V_{\beta+1}}(\boldsymbol{K}_\alpha), \bigcap_{\gamma \le \beta} \mathcal{Y}_{\alpha,\gamma}\right) \enspace .$$
- For any limit ordinal $\beta < \nu$, $\mathcal{Y}_{\alpha,\beta} = \mathcal{I}_\mathcal{L}$ 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 148, $\mathcal{Y}_{\alpha,\beta}$ is saturated relative to B_β. Thus, by Observations 100(2) and 95(4),
$$\mathcal{Y}_{\alpha,\beta} = \sigma(\mathcal{X}_\alpha, B_\beta) = \sigma(\sigma(\mathcal{M}, A_\alpha), B_\beta) = \sigma(\mathcal{M}, A_\alpha \cap B_\beta) \enspace .$$

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 $\beta + 1 < \nu$,
$$\boldsymbol{K}'_{\beta+1} = e_{V_\beta}\left(b_{V_{\beta+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, V_\beta)\right) \enspace .$$
- 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 $\alpha + 1 < \mu$,
$$\boldsymbol{K}_{0,\beta} = b_{V_0}(\boldsymbol{K}'_\beta) \enspace , \tag{D.8}$$
$$\boldsymbol{K}_{\alpha+1,\beta} = e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}'_\beta), \sigma(\mathcal{M}, U_\alpha)\right) \enspace . \tag{D.9}$$

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 150 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 150 and 151, achieving the following result:
$$\begin{aligned}
\boldsymbol{K}_{\alpha+1,0} &= b_{V_0}(\boldsymbol{K}_{\alpha+1}) \\
&= b_{V_0}(e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha))) \\
&= e_{U_\alpha}(b_{V_0}(b_{U_{\alpha+1}}(\boldsymbol{K})), \sigma(\mathcal{M}, U_\alpha)) \\
&= e_{U_\alpha}(b_{U_{\alpha+1} \cap V_0}(\boldsymbol{K}), \sigma(\mathcal{M}, U_\alpha)) \\
&= e_{U_\alpha}(b_{U_{\alpha+1}}(b_{V_0}(\boldsymbol{K})), \sigma(\mathcal{M}, U_\alpha)) \\
&= e_{U_\alpha}(b_{U_{\alpha+1}}(\boldsymbol{K}'_0), \sigma(\mathcal{M}, U_\alpha)) \enspace .
\end{aligned}$$
Now suppose that β is an ordinal such that $\beta + 1 < \nu$. Using Lemmas 150 and 151 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 150, 151 and 153 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 154 to replace $\sigma(\mathcal{M}, U_\alpha)$ in (D.9) by $\sigma(\mathcal{M}, U_\alpha \cap B_\beta)$. Furthermore, by consecutively using Observation 95(4), Lemmas 155 and 156 and Observation 95(4) 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, (D.9) 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, (D.8) and (D.9) 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 147 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 149 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 156 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 $\mathcal{Z}_\beta = \sigma(\mathcal{M}, B_\beta) = \sigma(\mathcal{M}, \emptyset) = \mathfrak{I}_{\mathcal{L}}$.

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 Observation 100(1) that

$$\mathcal{M} = \bigcap_{\beta<\nu} \sigma(\mathcal{M}, B_\beta) = \bigcap_{\beta<\nu} \mathcal{Z}_\beta \ .$$

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. □

D.3. Properties

Theorem 76 *(Faithfulness w.r.t. MKNF knowledge bases). Suppose that* S *is faithful to the stable model 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 54 and Proposition 15. □

Theorem 77 *(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 72, $\mathcal{M}$ 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 78 *(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 LP 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 72, $\mathcal{M}$ 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 79 *(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

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\beta\le\alpha} \mathcal{X}_\beta\right) \ .$$

- For any limit ordinal α, $\mathcal{X}_\alpha = \mathfrak{I}_{\mathcal{L}}$.

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 148 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

$$\mathcal{X}_{\alpha+1} \models e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}_{n-1}), \textstyle\bigcap_{\beta\le\alpha} \mathcal{X}_\beta\right) \ .$$

- For any limit ordinal α, $\mathcal{X}_\alpha = \mathfrak{I}_{\mathcal{L}}$ is saturated relative to $A_\alpha = \emptyset$.

Thus, by Observation 100(1), $\mathcal{M}$ is sequence-saturated relative to $\boldsymbol{A}$, by Observation 100(2), $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$, and by Lemma 156, $\bigcap_{\beta\le\alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$.

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 $\mathrm{pr}(\phi) \subseteq A_\alpha$. Due to the above considerations we can then conclude that $\mathcal{X}_\alpha \models \phi$. Furthermore,

$$\mathcal{X}_\alpha \models \phi \Longleftrightarrow \sigma(\mathcal{M}, A_\alpha) \models \phi \Longleftrightarrow \sigma(\mathcal{M}, A_\alpha)^{[A_\alpha]} \models \phi \Longleftrightarrow \mathcal{M}^{[A_\alpha]} \models \phi \Longleftrightarrow \mathcal{M} \models \phi \ .$$

On the other hand, if $\phi = \kappa(\pi)$ for some rule π, then the there exists a unique non-limit ordinal α such that $\mathrm{pr}(\mathsf{H}_\pi) \subseteq A_\alpha$. Suppose that $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$. If $\alpha = 0$, then it follows that $\pi \in b_{U_0}(\mathcal{K}_{n-1})$ and it follows that $\mathcal{X}_0 \models \kappa(\mathsf{B}_\pi)$. Consequently, $\mathcal{X}_0 \models \kappa(\mathsf{H}_\pi)$ and we can conclude that $\mathcal{M} \models \kappa(\mathsf{H}_\pi)$. If $\alpha = \beta + 1$, then there is a rule $\sigma \in e_{U_\beta}(b_{U_{\beta+1}}(\mathcal{K}_{n-1}), \bigcap_{\gamma\le\beta} \mathcal{X}_\gamma)$ and $\mathcal{X}_{\beta+1} \models \kappa(\mathsf{B}_\sigma)$, Consequently, $\mathcal{X}_{\beta+1} \models \kappa(\mathsf{H}_\sigma)$ and we obtain $\mathcal{M} \models \kappa(\mathsf{H}_\sigma)$. In either case, $\mathcal{M} \models \phi$. □

Definition 157 *(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 LP interpretation J such that $\mathsf{pr}(J) \subseteq \mathsf{pr}(\mathcal{P}_0)$ and $J \models \mathcal{P}_0$.

Definition 158 *(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
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{X}_\beta\right) \; .$$
3. For any limit ordinal α, $\mathcal{X}_\alpha = \mathfrak{I}_\mathcal{L}$.
4. $\bigcap_{\alpha<\mu} \mathcal{X}_\alpha \neq \emptyset$.

Remark 159. For the remainder of this section we assume that $\diamond$ has the Abstract Splitting Sequence Property, conserves the language, respects fact update and satisfies (FO2.⊤) as well as (FO8.2), and S is faithful to the stable model semantics.

Proposition 160. *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 model semantics and for every stable model J or a program $\mathcal{P}$, $J \models \mathcal{P}$. □

Proposition 161. *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
$$\mathcal{Y}_0 = \left[\!\!\left[\diamond \left\langle \kappa(b_{U_0}(\mathcal{O}_i)) \cup \left\{ \, l \mid (l.) \in b_{U_0}(\mathcal{P}_i) \, \right\} \right\rangle_{i<n} \right]\!\!\right] \; .$$
Otherwise, $\mathcal{Y}_0$ corresponds to the least set of objective literals J such that
$$J \models b_{U_0}(\mathcal{P}_0) \; .$$
- For any ordinal α such that $\alpha + 1 < \mu$, if
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{Y}_\beta\right)$$
is ontology-based, then
$$\mathcal{Y}_{\alpha+1} = \left[\!\!\left[\diamond \left\langle \kappa(t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i))) \cup \left\{ \, l \, \middle| \, (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{Y}_\beta\right) \, \right\} \right\rangle_{i<n} \right]\!\!\right] \; .$$
Otherwise, $\mathcal{Y}_{\alpha+1}$ corresponds to the least set of objective literals J such that
$$J \models e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_0), \textstyle\bigcap_{\beta\leq\alpha} \mathcal{Y}_\beta\right) \; .$$
- For any limit ordinal α, $\mathcal{Y}_\alpha = \mathfrak{I}_\mathcal{L}$.

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 LP 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
$$\mathcal{X}_0 = \{ \, I \in \mathfrak{I}_\mathcal{L} \mid I \models J \, \} \subseteq \left\{ \, I \in \mathfrak{I}_\mathcal{L} \mid I \models J' \, \right\} = \mathcal{Y}_0 \; .$$

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,

$$e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{X}) \supseteq e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{Y}) \ . \tag{D.10}$$

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 (D.10), 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 LP 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 (D.10) that $J \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \mathcal{Y})$, so $J \supseteq J'$ and we can conclude that

$$\mathcal{X}_{\alpha+1} = \{ I \in \mathfrak{I}_\mathcal{L} \mid I \models J \} \subseteq \left\{ I \in \mathfrak{I}_\mathcal{L} \mid I \models J' \right\} = \mathcal{Y}_{\alpha+1} \ .$$

3° The case when α is a limit ordinal follows trivially from $\mathcal{X}_\alpha = \mathcal{Y}_\alpha = \mathfrak{I}_\mathcal{L}$.

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 model semantics that $\boldsymbol{Y}$ is a solution to $\boldsymbol{K}$ w.r.t. $\boldsymbol{U}$. □

Corollary 162. *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 160 and 161. □

Proposition 163. *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 $\boldsymbol{T} = \langle T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0), \kappa(\mathcal{O}_1), \ldots, \kappa(\mathcal{O}_{n-1}) \rangle$ where

$$T_{\mathcal{P}_0}(\mathcal{M}) = \bigcup \{ \mathsf{H}_\pi \mid \pi \in \mathcal{P}_0 \wedge \mathcal{M} \models \kappa(\mathsf{B}_\pi) \} \ .$$

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 Abstract 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 Observation 100(2) and Lemma 156,

$$\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha) \qquad \text{and} \qquad \bigcap_{\beta\leq\alpha} \mathcal{X}_\alpha = \sigma(\mathcal{M}, U_\alpha) \ .$$

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

$$\boldsymbol{T}' = \langle \kappa(b_{U_0}(\mathcal{O}_i)) \cup \{ l \mid (l.) \in b_{U_0}(\mathcal{P}_i) \} \rangle_{i<n} \ .$$

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 LP interpretation J such that $J \models b_{U_0}(\mathcal{P}_0)$, then by (FO2.⊤),

$$[\![\diamond b_{A_0}(\boldsymbol{T})]\!] = [\![\diamond \langle b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \ldots, \emptyset \rangle]\!] = [\![b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \ .$$

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 $\mathsf{H}_\pi = l$, $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$ and $\mathrm{pr}(l) \subseteq U_0$. Since U_0 is a splitting set for $\mathcal{P}_0$, $\mathrm{pr}(\mathsf{B}_\pi) \subseteq U_0$. Consequently, $\mathcal{X}_0 = \sigma(\mathcal{M}, U_0) \models \kappa(\mathsf{B}_\pi)$ and we obtain that $J \models \mathsf{B}_\pi$. 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

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right)$$

is ontology-based and $\mathcal{X}_{\alpha+1} = [\![\diamond \boldsymbol{T}']\!]$ where

$$\boldsymbol{T}' = \left\langle \kappa(t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i))) \cup \left\{ l \;\middle|\; (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right) \right\} \right\rangle_{i<n} \enspace .$$

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 $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta)$ contains only facts, every rule $\pi \in \mathcal{P}_0$ such that $\mathrm{pr}(\mathsf{H}_\pi) \subseteq U_{\alpha+1}$ satisfies $\mathrm{pr}(\mathsf{B}_\pi) \subseteq U_\alpha$ and due to the fact that $\bigcap_{\beta \le \alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$ we can conclude that

$$b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})) = \left\{ l \;\middle|\; (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right) \right\} \enspace .$$

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

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right)$$

is rule-based and $\mathcal{X}_{\alpha+1}$ corresponds to some LP interpretation J such that $J \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta)$, then by (FO2.$\top$),

$$[\![\diamond\, b_{A_{\alpha+1}}(\boldsymbol{T})]\!] = [\![\diamond \langle b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \dots, \emptyset \rangle]\!] = [\![b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \enspace .$$

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 $\mathsf{H}_\pi = l$, $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$ and $\mathrm{pr}(l) \subseteq A_{\alpha+1}$. It follows that there is some rule $\sigma \in e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta)$ with $\mathsf{H}_\sigma = l$, $\mathcal{M} \models \kappa(\mathsf{B}_\sigma)$ and $\mathrm{pr}(\sigma) \subseteq A_{\alpha+1}$. Consequently, $\mathcal{X}_{\alpha+1} = \sigma(\mathcal{M}, A_{\alpha+1}) \models \kappa(\mathsf{B}_\sigma)$ and so $J \models \mathsf{B}_\sigma$. 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.$\top$) that

$$\mathcal{X}_\alpha = \mathfrak{I}_\mathcal{L} = [\![\emptyset]\!] = [\![\diamond \langle \emptyset, \emptyset, \dots, \emptyset \rangle]\!] = [\![\diamond\, b_\emptyset(\boldsymbol{T})]\!] = [\![\diamond\, b_{A_\alpha}(\boldsymbol{T})]\!] \enspace . \qquad \square$$

Proposition 164. *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 $\boldsymbol{T} = \langle T_{\mathcal{P}_0}(\mathcal{M}) \cup \kappa(\mathcal{O}_0), \kappa(\mathcal{O}_1), \dots, \kappa(\mathcal{O}_{n-1}) \rangle$ where

$$T_{\mathcal{P}_0}(\mathcal{M}) = \bigcup \{ \mathsf{H}_\pi \mid \pi \in \mathcal{P}_0 \wedge \mathcal{M} \models \kappa(\mathsf{B}_\pi) \} \enspace .$$

By the definition of $T^\diamond_{\boldsymbol{K}}$, $T^\diamond_{\boldsymbol{K}}(\mathcal{M}) = [\![\diamond \boldsymbol{T}]\!]$. By the Abstract Splitting Sequence Property of $\diamond$ it follows that

$$\mathcal{M} = T^\diamond_{\boldsymbol{K}}(\mathcal{M}) = [\![\diamond \boldsymbol{T}]\!] = \bigcap_{\alpha<\mu} [\![\diamond\, b_{A_\alpha}(\boldsymbol{T})]\!] \enspace .$$

Let $\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha)$ for all $\alpha < \mu$. Since $\diamond$ conserves the language, it follows from Observation 100(2) that for all $\alpha < \mu$,

$$\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha) = [\![\diamond\, b_{A_\alpha}(\boldsymbol{T})]\!] \enspace .$$

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

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right) \enspace .$$

3. For any limit ordinal α, $\mathcal{X}_\alpha = \mathfrak{I}_\mathcal{L}$.
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

$$\boldsymbol{T}' = \langle \kappa(b_{U_0}(\mathcal{O}_i)) \cup \{\, l \mid (l.) \in b_{U_0}(\mathcal{P}_i) \,\}\rangle_{i<n} \enspace .$$

Since $\mathcal{P}_i = \emptyset$ for all $i > 0$, it follows that $b_{U_0}(\boldsymbol{T}) = \boldsymbol{T}'$ and thus

$$\mathcal{X}_0 = [\![\diamond b_{A_0}(\boldsymbol{T})]\!] = [\![\diamond b_{U_0}(\boldsymbol{T})]\!] = [\![\diamond \boldsymbol{T}']\!] \enspace .$$

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 LP interpretation J such that $J \models b_{U_0}(\mathcal{P}_0)$. Note that due to (FO2.⊤),

$$\mathcal{X}_0 = [\![\diamond b_{A_0}(\boldsymbol{T})]\!] = [\![\diamond \langle b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \dots, \emptyset\rangle]\!] = [\![b_{U_0}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \enspace .$$

Put $J = \{\, l \in \mathit{Lits}_{\mathsf{G}} \mid \mathcal{X}_0 \models l \,\}$. It follows that if $J \models \mathsf{B}_\pi$ for some rule $\pi \in b_{U_0}(\mathcal{P}_0)$, then $\mathcal{X}_0 \models \kappa(\mathsf{B}_\pi)$ and thus $\mathcal{M} \models \kappa(\mathsf{B}_\pi)$. Consequently, $\mathsf{H}_\pi \in T_{\mathcal{P}_0}(\mathcal{M})$ and we conclude that $\mathcal{X}_0 \models \mathsf{H}_\pi$, thus also $J \models \mathsf{H}_\pi$.

2. For a non-limit ordinal $\alpha + 1$ we have $A_{\alpha+1} = U_{\alpha+1} \setminus U_\alpha$. Suppose first that

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right)$$

is ontology-based. We have to prove that $\mathcal{X}_{\alpha+1} = [\![\diamond \boldsymbol{T}']\!]$ where

$$\boldsymbol{T}' = \left\langle \kappa(t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{O}_i))) \cup \left\{\, l \,\middle|\, (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right) \right\}\right\rangle_{i<n} \enspace .$$

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 $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta)$ contains only facts, every rule $\pi \in \mathcal{P}_0$ such that $\mathrm{pr}(\mathsf{H}_\pi) \subseteq U_{\alpha+1}$ satisfies $\mathrm{pr}(\mathsf{B}_\pi) \subseteq U_\alpha$ and due to the fact that $\bigcap_{\beta \le \alpha} \mathcal{X}_\beta = \sigma(\mathcal{M}, U_\alpha)$ we can conclude that

$$b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})) = \left\{\, l \,\middle|\, (l.) \in e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{P}_i), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right) \right\} \enspace .$$

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

$$\mathcal{X}_{\alpha+1} = [\![\diamond b_{A_0}(\boldsymbol{T})]\!] = [\![\diamond \boldsymbol{T}']\!] \enspace .$$

On the other hand, if

$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\boldsymbol{K}), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta\right)$$

is rule-based, we need to prove that $\mathcal{X}_{\alpha+1}$ corresponds to some LP interpretation J such that $J \models e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta)$. Due to (FO2.⊤),

$$\mathcal{X}_{\alpha+1} = [\![\diamond b_{A_{\alpha+1}}(\boldsymbol{T})]\!] = [\![\diamond \left\langle b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M})), \emptyset, \dots, \emptyset\right\rangle]\!] = [\![b_{A_{\alpha+1}}(T_{\mathcal{P}_0}(\mathcal{M}))]\!] \enspace .$$

Put $J = \{\, l \in \mathit{Lits}_{\mathsf{G}} \mid \mathcal{X}_{\alpha+1} \models l \,\}$. It follows that if $J \models \mathsf{B}_\pi$ for some rule $\pi \in e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{P}_0), \bigcap_{\beta \le \alpha} \mathcal{X}_\beta)$, then there is a rule $\sigma \in b_{U_{\alpha+1}}(\mathcal{P}_0)$ such that $\mathsf{H}_\pi = \mathsf{H}_\sigma$ and $\mathcal{M} \models \kappa(\mathsf{B}_\sigma)$. Consequently, $\mathsf{H}_\pi \in T_{\mathcal{P}_0}(\mathcal{M})$ and we obtain $\mathcal{X}_{\alpha+1} \models \mathsf{H}_\pi$, thus also $J \models \mathsf{H}_\pi$.

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

$$\mathcal{X}_\alpha = \sigma(\mathcal{M}, A_\alpha) = \sigma(\mathcal{M}, \emptyset) = \mathfrak{I}_{\mathcal{L}} \enspace .$$

4. Since $\mathcal{M}$ is an MKNF interpretation, it follows that $\emptyset \neq \mathcal{M} = \bigcap_{\alpha < \mu} \mathcal{X}_\alpha$. □

Proposition 165. *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 $(\diamond, \mathsf{S})$-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 164 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 162, 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 160 and 163 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 160 and 163, $\mathcal{N} \subseteq T^{\diamond}_{\boldsymbol{K}}(\mathcal{N})$ 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 164 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 162, 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

$$\mathcal{M} = \bigcap_{\alpha<\mu} \sigma(\mathcal{M}, A_\alpha) \subseteq \bigcap_{\alpha<\mu} \mathcal{X}_\alpha = \mathcal{N} \ . \qquad \square$$

Proposition 166. *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 model semantics and by the definition of a stable model. $\square$

Theorem 80 *(Compatibility with update semantics from Section 3). Suppose that $\diamond$ satisfies (FO2.$\top$) and (FO8.2) and that S is faithful to the stable model 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 165 and 166. $\square$

D.4. Complexity of updating layered DMKBs

Theorem 88. *Let $\diamond$ be a first update operator and S a rule update semantics such that both $\diamond$ and S have the splitting sequence property, conserve the language and respect fact update. If query answering for $\diamond$ belongs to Ω_1 and query answering for S belongs to Ω_2, then hybrid query answering for $\diamond$ and S belongs to $\mathbf{NP}^{\Omega_1 \cup \Omega_2}$.*

Proof (sketch). Take a layered DMKB $\boldsymbol{K}$. We need to prove that the complexity class $\mathbf{NP}^{\Omega_1 \cup \Omega_2}$ contains the problem of deciding whether for some $(\diamond, \mathsf{S})$-dynamic MKNF model $\mathcal{M}$, $\mathcal{M} \models \kappa(S)$.

Since $\boldsymbol{K}$ is layered, some layering splitting sequence $\boldsymbol{U} = \langle U_i\rangle_{i<n}$ for $\boldsymbol{K}$ exists.[26] We define the sets of literals S_i for all $i < n$ as follows:

- $S_0 = \emptyset$,
- for any $i \in \mathbb{N}$ such that $i + 1 < n$,

$$S_{i+1} = \{\, L \in \mathsf{B}_\pi \mid \pi \in t_{U_i}(b_{U_{i+1}}(\boldsymbol{K})) \wedge \mathsf{pr}(L) \subseteq U_i \,\} \ .$$

Intuitively, the set S_i contains those body literals of rules within the i-th layer of $\boldsymbol{K}$ that are determined by one of the previous layers. For each $i < n$, we guess a set of literals $S_i^{\top} \subseteq S_i$, then we check that the guess determines some $(\diamond, \mathsf{S})$-dynamic MKNF model $\mathcal{M}$ of $\boldsymbol{K}$ such that $\mathcal{M} \models \kappa(S)$. In other words, we guess which literals from S_i are true in such $\mathcal{M}$.

The rest can be verified in deterministic polynomial time using our $(\Omega_1 \cup \Omega_2)$ oracle as follows. For each $i < n$, we determine the DMKB $\boldsymbol{K}_i$ and the set of literals S'_i as follows:

[26] Since all components of $\boldsymbol{K}$ are finite, we assume without loss of generality that $\boldsymbol{U}$ is finite.

- $\boldsymbol{K}_0 = b_{U_0}(\boldsymbol{K})$ and $S'_i = \{ L \in S \mid \mathsf{pr}(L) \subseteq U_0 \}$.
- For any $i \in \mathbb{N}$ such that $i + 1 < n$, $\boldsymbol{K}_{i+1}$ is obtained from $t_{U_i}(b_{U_{i+1}}(\boldsymbol{K}))$ by replacing each literal from S_i by $\top$ if it belongs to $S_i^\top$ and by $\bot$ otherwise. Furthermore,

$$S'_{i+1} = \{ L \in S \mid \mathsf{pr}(L) \subseteq U_{i+1} \setminus U_i \} \enspace .$$

Since $\boldsymbol{K}_i$ is a basic DMKB, it can be updated using either $\diamond$ of S alone. Thus, using the oracle we can verify that for each $i < n$,

$$\boldsymbol{K}_i \models \kappa(S_i \cup S'_i) \enspace .$$

Together with Definition 73, this leads to the conclusion that for some $(\diamond, \mathsf{S})$-dynamic MKNF model $\mathcal{M}$ of $\boldsymbol{K}$, $\mathcal{M} \models \kappa(S)$. □

References

[1] M. Alberti, A.S. Gomes, R. Gonçalves, J. Leite, M. Slota, Normative systems represented as hybrid knowledge bases, in: J. Leite, P. Torroni, T. Ågotnes, G. Boella, L. van der Torre (Eds.), Computational Logic in Multi-Agent Systems – Proceedings of the 12th International Workshop, CLIMA XII, Barcelona, Spain, July 17–18, 2011, in: Lecture Notes in Computer Science, vol. 6814, Springer, 2011, pp. 330–346.

[2] M. Alberti, M. Knorr, A.S. Gomes, J. Leite, R. Gonçalves, M. Slota, Normative systems require hybrid knowledge bases, in: W. van der Hoek, L. Padgham, V. Conitzer, M. Winikoff (Eds.), International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4–8, 2012 (3 Volumes), IFAAMAS, 2012, pp. 1425–1426.

[3] C.E. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: partial meet contraction and revision functions, J. Symb. Log. 50 (2) (1985) 510–530.

[4] J.J. Alferes, L.M. Pereira, Update-programs can update programs, in: J. Dix, L.M. Pereira, T.C. Przymusinski (Eds.), NMELP '96 Selected Papers from the Non-monotonic Extensions of Logic Programming, Bad Honnef, Germany, September 5–6, 1996, in: Lecture Notes in Computer Science, vol. 1216, Springer, ISBN 3-540-62843-6, 1996, pp. 110–131.

[5] J.J. Alferes, J.A. Leite, L.M. Pereira, H. Przymusinska, T.C. Przymusinski, Dynamic logic programming, in: A.G. Cohn, L.K. Schubert, S.C. Shapiro (Eds.), Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning, KR'98, Trento, Italy, June 2–5, 1998, Morgan Kaufmann, 1998, pp. 98–111.

[6] J.J. Alferes, J.A. Leite, L.M. Pereira, H. Przymusinska, T.C. Przymusinski, Dynamic updates of non-monotonic knowledge bases, J. Log. Program. 45 (1–3) (September/October 2000) 43–70.

[7] J.J. Alferes, F. Banti, A. Brogi, J.A. Leite, The refined extension principle for semantics of dynamic logic programming, Stud. Log. 79 (1) (2005) 7–32.

[8] J.J. Alferes, M. Knorr, T. Swift, Query-driven procedures for hybrid MKNF knowledge bases, ACM Trans. Comput. Log. 14 (2) (2013) 16.

[9] H. Andréka, J. van Bentham, I. Németi, Modal languages and bounded fragments of predicate logic, J. Philos. Log. 27 (1998) 217–274.

[10] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: L.P. Kaelbling, A. Saffiotti (Eds.), Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-05, Edinburgh, Scotland, UK, July 30–August 5, 2005, Professional Book Center, ISBN 0938075934, 2005, pp. 364–369.

[11] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, 2nd edition, Cambridge University Press, 2007.

[12] J. Babb, J. Lee, Module theorem for the general theory of stable models, Theory Pract. Log. Program. 12 (4–5) (2012) 719–735.

[13] F. Banti, J.J. Alferes, A. Brogi, P. Hitzler, The well supported semantics for multidimensional dynamic logic programs, in: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Proceedings of the 8th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 2005, Diamante, Italy, September 5–8, 2005, in: Lecture Notes in Computer Science, vol. 3662, Springer, ISBN 3-540-28538-5, 2005, pp. 356–368.

[14] P. Bonatti, Reasoning with infinite stable models, Artif. Intell. 156 (2004) 75–111.

[15] Y. Bong, The description logic ABox update problem revisited, Master's thesis, Dresden University of Technology, Dresden, Germany, February 2007.

[16] G. Brewka, Towards reactive multi-context systems, in: P. Cabalar, T.C. Son (Eds.), Logic Programming and Nonmonotonic Reasoning, 12th International Conference, LPNMR 2013, in: Lecture Notes in Computer Science, vol. 8148, Springer, 2013, pp. 1–10.

[17] G. Brewka, T. Eiter, Equilibria in heterogeneous nonmonotonic multi-context systems, in: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, July 22–26, 2007, AAAI Press, ISBN 978-1-57735-323-2, 2007, pp. 385–390.

[18] G. Brewka, T. Eiter, M. Fink, A. Weinzierl, Managed multi-context systems, in: T. Walsh (Ed.), IJCAI, IJCAI/AAAI, ISBN 978-1-57735-516-8, 2011, pp. 786–791.

[19] G. Brewka, S. Ellmauthaler, J. Pührer, Multi-context systems for reactive reasoning in dynamic environments, in: T. Schaub, G. Friedrich, B. O'Sullivan (Eds.), ECAI 2014 – 21st European Conference on Artificial Intelligence, in: Frontiers in Artificial Intelligence and Applications, vol. 263, IOS Press, 2014, pp. 159–164.

[20] F. Buccafurri, W. Faber, N. Leone, Disjunctive logic programs with inheritance, in: D.D. Schreye (Ed.), Proceedings of the 1999 International Conference on Logic Programming, ICLP 1999, Las Cruces, New Mexico, USA, November 29–December 4, 1999, The MIT Press, ISBN 0-262-54104-1, 1999, pp. 79–93.

[21] D. Calvanese, E. Kharlamov, W. Nutt, D. Zheleznyakov, Evolution of DL-Lite knowledge bases, in: P.F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J.Z. Pan, I. Horrocks, B. Glimm (Eds.), International Semantic Web Conference (1), Shanghai, China, November 7–11, 2010, in: Lecture Notes in Computer Science, vol. 6496, Springer, ISBN 978-3-642-17745-3, 2010, pp. 112–128.

[22] A. Colmerauer, H. Kanoui, P. Roussel, R. Pasero, Un systéme de communication homme-machine en français, Technical report, Groupe de Recherche en Intelligence Artificielle, Université d'Aix-Marseille II, 1973.

[23] N. Costa, M. Knorr, J. Leite, Next step for NoHR: OWL 2 QL, in: M. Arenas, O. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, S. Staab (Eds.), The Semantic Web – ISWC 2015 – 14th International Semantic Web Conference, Bethlehem, Pennsylvania, United States, October 11–15, 2015, in: Lecture Notes in Computer Science, Springer, 2015.

[24] M. Dalal, Investigations into a theory of knowledge base revision, in: Proceedings of the 7th National Conference on Artificial Intelligence, AAAI 1988, St. Paul, MN, USA, August 21–26, 1988, AAAI Press/The MIT Press, ISBN 0-262-51055-3, 1988, pp. 475–479.

[25] J. de Bruijn, D. Pearce, A. Polleres, A. Valverde, A semantical framework for hybrid knowledge bases, Knowl. Inf. Syst. 25 (1) (2010) 81–104.

[26] J. de Bruijn, T. Eiter, A. Polleres, H. Tompits, Embedding nonground logic programs into autoepistemic logic for knowledge-base combination, ACM Trans. Comput. Log. 12 (3) (2011) 20.

[27] G. De Giacomo, M. Lenzerini, A. Poggi, R. 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, 2006.

[28] G. De Giacomo, M. Lenzerini, A. Poggi, R. 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, Vancouver, British Columbia, Canada, July 22–26, 2007, AAAI Press, ISBN 978-1-57735-323-2, 2007, pp. 403–408.
[29] G. De Giacomo, M. Lenzerini, A. Poggi, R. Rosati, On instance-level update and erasure in description logic ontologies, J. Log. Comput. 19 (5) (2009) 745–770.
[30] J.P. Delgrande, A program-level approach to revising logic programs under the answer set semantics, in: 26th Int'l. Conference on Logic Programming (ICLP'10), Theory Pract. Log. Program. 10 (4–6) (July 2010) 565–580 (Special Issue).
[31] J.P. Delgrande, T. Schaub, H. Tompits, A preference-based framework for updating logic programs, in: C. Baral, G. Brewka, J.S. Schlipf (Eds.), Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 2007, Tempe, AZ, USA, May 15–17, 2007, in: Lecture Notes in Computer Science, vol. 4483, Springer, ISBN 978-3-540-72199-4, 2007, pp. 71–83.
[32] J.P. Delgrande, T. Schaub, H. Tompits, S. Woltran, Belief revision of logic programs under answer set semantics, in: G. Brewka, J. Lang (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR'08, AAAI Press, 2008, pp. 411–421.
[33] J.P. Delgrande, T. Schaub, H. Tompits, S. Woltran, A model-theoretic approach to belief change in answer set programming, ACM Trans. Comput. Log. 14 (2) (2013) 14, http://dx.doi.org/10.1145/2480759.2480766.
[34] J. Dix, A classification theory of semantics of normal logic programs: I. Strong properties, Fundam. Inform. 22 (3) (1995) 227–255.
[35] P. Doherty, W. Lukaszewicz, E. Madalinska-Bugaj, The PMA and relativizing minimal change for action update, in: A.G. Cohn, L.K. Schubert, S.C. Shapiro (Eds.), Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning, KR'98, Trento, Italy, June 2–5, 1998, Morgan Kaufmann, 1998, pp. 258–269.
[36] C. Drescher, H. Liu, F. Baader, P. Steinke, M. Thielscher, Putting ABox updates into action, in: Proceedings of the 8th IJCAI International Workshop on Nonmontonic Reasoning, Action and Change, NRAC-09, 2009.
[37] T. Eiter, M. Fink, G. Sabbatini, H. Tompits, On properties of update sequences based on causal rejection, Theory Pract. Log. Program. 2 (6) (2002) 721–777.
[38] T. Eiter, G. Ianni, R. Schindlauer, H. Tompits, A uniform integration of higher-order reasoning and external evaluations in answer-set programming, in: L.P. Kaelbling, A. Saffiotti (Eds.), Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-05, Edinburgh, Scotland, UK, July 30–August 5, 2005, Professional Book Center, ISBN 0938075934, 2005, pp. 90–96.
[39] K.D. Forbus, Introducing actions into qualitative simulation, in: N.S. Sridharan (Ed.), Proceedings of the 11th International Joint Conference on Artificial Intelligence, IJCAI'89, Detroit, MI, USA, August 1989, Morgan Kaufmann, ISBN 1-55860-094-9, 1989, pp. 1273–1278.
[40] P. Gärdenfors, Belief revision: an introduction, in: Belief Revision, Cambridge University Press, 1992, pp. 1–28.
[41] A.V. Gelder, K.A. Ross, J.S. Schlipf, The well-founded semantics for general logic programs, J. ACM 38 (3) (1991) 620–650.
[42] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: R.A. Kowalski, K.A. Bowen (Eds.), Proceedings of the 5th International Conference and Symposium on Logic Programming, ICLP/SLP 1988, Seattle, Washington, August 15–19, 1988, MIT Press, 1988, pp. 1070–1080.
[43] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Gener. Comput. 9 (3–4) (1991) 365–385.
[44] S. Gomes, J. Alferes, T. Swift, A goal-directed implementation of query answering for hybrid MKNF knowledge bases, Theory Pract. Log. Program. 14 (2) (2014) 239–264.
[45] R. Gonçalves, M. Knorr, J. Leite, Evolving multi-context systems, in: T. Schaub, G. Friedrich, B. O'Sullivan (Eds.), ECAI 2014 – 21st European Conference on Artificial Intelligence, in: Frontiers in Artificial Intelligence and Applications, vol. 263, IOS Press, 2014, pp. 375–380.
[46] S.O. Hansson, Reversing the Levi identity, J. Philos. Log. 22 (6) (December 1993) 637–669.
[47] S.J. Hegner, Specification and implementation of programs for updating incomplete information databases, in: M.Y. Vardi (Ed.), Proceedings of the Sixth ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems, PODS'87, ACM, ISBN 0-89791-223-3, 1987, pp. 146–158.
[48] A. Herzig, The PMA revisited, in: L.C. Aiello, J. Doyle, S.C. Shapiro (Eds.), Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning, KR'96, Cambridge, Massachusetts, USA, November 5–8, 1996, Morgan Kaufmann, ISBN 1-55860-421-9, 1996, pp. 40–50.
[49] A. Herzig, On updates with integrity constraints, in: J.P. Delgrande, J. Lang, H. Rott, J.-M. Tallon (Eds.), Belief Change in Rational Agents: Perspectives from Artificial Intelligence, Philosophy, and Economics, August 7–12, 2005, in: Dagstuhl Seminar Proceedings, vol. 05321, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.
[50] A. Herzig, O. Rifi, Update operations: a review, in: H. Prade (Ed.), Proceedings of the 13th European Conference on Artificial Intelligence, ECAI'98, Brighton, UK, August 23–28, 1998, John Wiley and Sons, 1998, pp. 13–17.
[51] A. Herzig, O. Rifi, Propositional belief base update and minimal change, Artif. Intell. 115 (1) (1999) 107–138.
[52] P. Hitzler, B. Parsia, Ontologies and rules, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, second edition, in: International Handbooks on Information Systems, Springer, Berlin, ISBN 978-3-540-70999-2, 2009, pp. 111–132.
[53] V. Ivanov, M. Knorr, J. Leite, A query tool for $\mathcal{EL}$ with non-monotonic rules, in: H. Alani, L. Kagal, A. Fokoue, P.T. Groth, C. Biemann, J.X. Parreira, L. Aroyo, N.F. Noy, C. Welty, K. Janowicz (Eds.), Proceedings of the 12th International Semantic Web Conference (ISWC'13), Part I, in: Lecture Notes in Computer Science, vol. 8218, Springer, ISBN 978-3-642-41334-6, 2013, pp. 216–231.
[54] T. Janhunen, E. Oikarinen, H. Tompits, S. Woltran, Modularity aspects of disjunctive stable models, J. Artif. Intell. Res. 35 (2009) 813–857.
[55] T. Kaminski, M. Knorr, J. Leite, Efficient paraconsistent reasoning with ontologies and rules, in: Q. Yang, M. Wooldridge (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25–31, 2015, AAAI Press, 2015, pp. 3098–3105, http://ijcai.org/papers15/Abstracts/IJCAI15-437.html.
[56] H. Katsuno, A.O. Mendelzon, On the difference between updating a knowledge base and revising it, in: J.F. Allen, R. Fikes, E. Sandewall (Eds.), Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, KR'91, Cambridge, MA, USA, April 22–25, 1991, Morgan Kaufmann Publishers, ISBN 1-55860-165-1, 1991, pp. 387–394.
[57] A.M. Keller, M. Winslett, On the use of an extended relational model to handle changing incomplete information, IEEE Trans. Softw. Eng. 11 (7) (1985) 620–633.
[58] E. Kharlamov, D. Zheleznyakov, D. Calvanese, Capturing model-based ontology evolution at the instance level: the case of DL-Lite, J. Comput. Syst. Sci. 79 (6) (2013) 835–872.
[59] M. Knorr, J.J. Alferes, P. Hitzler, Local closed world reasoning with description logics under the well-founded semantics, Artif. Intell. 175 (9–10) (2011) 1528–1554.
[60] M. Knorr, J. Leite, M. Slota, M. Homola, What if no hybrid reasoner is available? Hybrid MKNF in multi-context systems, J. Log. Comput. (ISSN 1465-363X) (2013) 1–32, http://dx.doi.org/10.1093/logcom/ext062.
[61] R.A. Kowalski, Predicate logic as programming language, in: IFIP Congress, 1974, pp. 569–574.
[62] M. Krötzsch, S. Rudolph, P. Hitzler, Description logic rules, in: M. Ghallab, C.D. Spyropoulos, N. Fakotakis, N. Avouris (Eds.), Proceedings of the 18th European Conference on Artificial Intelligence, ECAI2008, Patras, Greece, July 2008, IOS Press, 2008, pp. 80–84.
[63] P. Krümpelmann, G. 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.
[64] J. Lee, R. Palla, Integrating rules and ontologies in the first-order stable model semantics (preliminary report), in: J.P. Delgrande, W. Faber (Eds.), Logic Programming and Nonmonotonic Reasoning – 11th International Conference, LPNMR 2011, in: Lecture Notes in Computer Science, vol. 6645, Springer, 2011, pp. 248–253.

[65] J.A. Leite, Evolving Knowledge Bases, Frontiers of Artificial Intelligence and Applications, vol. 81, IOS Press, ISBN 1-58603-278-X, 2003, xviii + 307 pp., Hardcover.
[66] J.A. Leite, L.M. Pereira, Generalizing updates: from models to programs, in: J. Dix, L.M. Pereira, T.C. Przymusinski (Eds.), Proceedings of the 3rd International Workshop on Logic Programming and Knowledge Representation, LPKR '97, Port Jefferson, New York, USA, October 17, 1997, in: Lecture Notes in Computer Science, vol. 1471, Springer, ISBN 3-540-64958-1, 1997, pp. 224–246.
[67] M. Lenzerini, D.F. Savo, On the evolution of the instance level of DL-Lite knowledge bases, in: R. Rosati, S. Rudolph, M. Zakharyaschev (Eds.), Proceedings of the 24th International Workshop on Description Logics, DL 2011, Barcelona, Spain, July 13–16, 2011, in: CEUR Workshop Proceedings, vol. 745, 2011, CEUR-WS.org.
[68] V. Lifschitz, Frames in the space of situations, Artif. Intell. 46 (3) (1990) 365–376.
[69] V. Lifschitz, Nonmonotonic databases and epistemic queries, in: Proceedings of the 12th International Joint Conference on Artificial Intelligence, IJCAI'91, 1991, pp. 381–386.
[70] V. Lifschitz, Minimal belief and negation as failure, Artif. Intell. 70 (1–2) (1994) 53–72.
[71] V. Lifschitz, H. Turner, Splitting a logic program, in: P.V. Hentenryck (Ed.), Proceedings of the 11th International Conference on Logic Programming, ICLP 1994, Santa Margherita Ligure, Italy, June 13–18, 1994, MIT Press, ISBN 0-262-72022-1, 1994, pp. 23–37.
[72] V. Lifschitz, D. Pearce, A. Valverde, Strongly equivalent logic programs, ACM Trans. Comput. Log. 2 (4) (2001) 526–541.
[73] H. Liu, C. Lutz, M. Miličić, F. Wolter, Updating description logic ABoxes, in: P. Doherty, J. Mylopoulos, C.A. Welty (Eds.), Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning, KR'06, Lake District of the United Kingdom, June 2–5, 2006, AAAI Press, ISBN 978-1-57735-271-6, 2006, pp. 46–56.
[74] J.W. Lloyd, Foundations of Logic Programming, 2nd edition, Springer, ISBN 3-540-18199-7, 1987.
[75] D. Makinson, General theory of cumulative inference, in: R. Michael, J. De Kleer, M.L. Ginsberg, E. Sandewall (Eds.), Proceedings of the 2nd International Workshop on Non-monotonic Reasoning, NMR'88, Grassau, FRG, June 13–15, 1988, in: Lecture Notes in Computer Science, vol. 346, Springer, ISBN 3-540-50701-9, 1988, pp. 1–18.
[76] V.W. Marek, M. Truszczynski, Revision programming, Theor. Comput. Sci. 190 (2) (1998) 241–277.
[77] B. Motik, R. Rosati, Reconciling description logics and rules, J. ACM 57 (5) (2010) 93–154.
[78] M. Osorio, V. Cuevas, Updates in answer set programming: an approach based on basic structural properties, Theory Pract. Log. Program. 7 (4) (2007) 451–479.
[79] C. Papadimitriou, Computational Complexity, Addison–Wesley, 1994.
[80] C. Sakama, K. Inoue, An abductive framework for computing knowledge base updates, Theory Pract. Log. Program. 3 (6) (2003) 671–713.
[81] D. Scott, A decision method for validity of sentences in two variables, J. Symb. Log. 27 (1962).
[82] J. Šefránek, Static and dynamic semantics: preliminary report, in: Mexican International Conference on Artificial Intelligence, 2011, pp. 36–42.
[83] M. Slota, J. Leite, Marrying stable models with belief update, in: Proceedings of the 13th International Workshop on Non-monotonic Reasoning, NMR 2010, 2010.
[84] M. Slota, J. Leite, Towards closed world reasoning in dynamic open worlds, in: Proceedings of the 26th International Conference on Logic Programming (ICLP'10), Theory Pract. Log. Program. 10 (4–6) (2010) 547–564 (Special Issue).
[85] M. Slota, J. Leite, On semantic update operators for answer-set programs, in: H. Coelho, R. Studer, M. Wooldridge (Eds.), Proceedings of the 19th European Conference on Artificial Intelligence, ECAI 2010, Lisbon, Portugal, August 16–20, 2010, in: Frontiers in Artificial Intelligence and Applications, vol. 215, IOS Press, ISBN 978-1-60750-605-8, 2010, pp. 957–962.
[86] M. Slota, J. Leite, Back and forth between rules and SE-models, in: J.P. Delgrande, W. Faber (Eds.), Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR-11, Vancouver, Canada, May 16–19, 2011, in: Lecture Notes in Computer Science, vol. 6645, Springer, ISBN 978-3-642-20894-2, 2011, pp. 174–186.
[87] M. Slota, J. Leite, Robust equivalence models for semantic updates of answer-set programs, in: G. Brewka, T. Eiter, S.A. McIlraith (Eds.), Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning, KR 2012, Rome, Italy, June 10–14, 2012, AAAI Press, ISBN 978-1-57735-560-1, 2012, pp. 158–168.
[88] M. Slota, J. Leite, A unifying perspective on knowledge updates, in: L.F. del Cerro, A. Herzig, J. Mengin (Eds.), Proceedings of the 13th European Conference on Logics in Artificial Intelligence, JELIA 2012, Toulouse, France, September 26–28, 2012, in: Logics in Artificial Intelligence (LNAI), vol. 7519, Springer, 2012, pp. 372–384.
[89] M. Slota, J. Leite, On condensing a sequence of updates in answer-set programming, in: F. Rossi (Ed.), Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13, IJCAI/AAAI, ISBN 978-1-57735-633-2, 2013, pp. 1097–1103.
[90] M. Slota, J. Leite, The rise and fall of semantic rule updates based on SE-models, Theory Pract. Log. Program. 14 (6) (2014) 869–907, http://dx.doi.org/10.1017/S1471068413000100.
[91] M. Slota, J. Leite, T. Swift, Splitting and updating hybrid knowledge bases, in: Proceedings of the 27th International Conference on Logic Programming (ICLP'11), Theory Pract. Log. Program. 11 (4–5) (2011) 801–819 (Special Issue).
[92] H. 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, Portland, Oregon, August 4–8, 1996, vol. 1, AAAI Press/The MIT Press, ISBN 0-262-51091-X, 1996, pp. 645–651.
[93] H. Turner, Strong equivalence made easy: nested expressions and weight constraints, Theory Pract. Log. Program. 3 (4–5) (2003) 609–622.
[94] Y. Wang, Z. Zhuang, K. Wang, Belief change in nonmonotonic multi-context systems, in: P. Cabalar, T.C. Son (Eds.), LPNMR, in: Lecture Notes in Computer Science, vol. 8148, Springer, ISBN 978-3-642-40563-1, 2013, pp. 543–555.
[95] M. Winslett, Reasoning about action using a possible models approach, in: Proceedings of the 7th National Conference on Artificial Intelligence, AAAI 1988, Saint Paul, MN, USA, August 21–26, 1988, AAAI Press/The MIT Press, ISBN 0-262-51055-3, 1988, pp. 89–93.
[96] M. Winslett, Updating Logical Databases, Cambridge University Press, New York, USA, ISBN 0-521-37371-9, 1990.
[97] Y. Zhang, Logic program-based updates, ACM Trans. Comput. Log. 7 (3) (2006) 421–472.
[98] Y. Zhang, N.Y. Foo, A unified framework for representing logic program updates, in: M.M. Veloso, S. Kambhampati (Eds.), Proceedings of the 20th National Conference on Artificial Intelligence, AAAI 2005, Pittsburgh, Pennsylvania, USA, July 9–13, 2005, AAAI Press/The MIT Press, ISBN 1-57735-236-X, 2005, pp. 707–713.