Splitting and Updating Hybrid Knowledge Bases

MARTIN SLOTA* and JOÃO LEITE and TERRANCE SWIFT
CENTRIA & Departamento de Informática
Universidade Nova de Lisboa
2829-516 Caparica, Portugal

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

Abstract

Over the years, nonmonotonic rules have proven to be a very expressive and useful knowledge representation paradigm. They have recently been used to complement the expressive power of Description Logics (DL), leading to the study of integrative formal frameworks, generally referred to as *hybrid knowledge bases*, where both DL axioms and rules can be used to represent knowledge. The need to use these hybrid knowledge bases in dynamic domains has called for the development of update operators, which, given the substantially different way Description Logics and rules are usually updated, turned out to be an extremely difficult task.

In (Slota and Leite 2010b), a first step towards addressing this problem was taken, and an update operator for hybrid knowledge bases was proposed. Despite its significance – not only for being the first update operator for hybrid knowledge bases in the literature, but also because it has some applications – this operator was defined for a restricted class of problems where only the ABox was allowed to change, which considerably diminished its applicability. Many applications that use hybrid knowledge bases in dynamic scenarios require both DL axioms and rules to be updated.

In this paper, motivated by real world applications, we introduce an update operator for a large class of hybrid knowledge bases where both the DL component as well as the rule component are allowed to dynamically change. We introduce splitting sequences and splitting theorem for hybrid knowledge bases, use them to define a modular update semantics, investigate its basic properties, and illustrate its use on a realistic example about cargo imports.

KEYWORDS: hybrid knowledge base, update, splitting theorem, ontology, logic program

1 Introduction

Increasingly many real world applications need to intelligently access and reason with large amounts of dynamically changing, structured and highly interconnected information. The family of Description Logics (DLs) (Baader et al. 2003), generally characterised as decidable fragments of first-order logic, have become the established standard for specifying and sharing concepts relevant to a particular domain of interest, more commonly referred to as *ontologies*. DLs can be seen as some of the most expressive formalisms based on Classical Logic for which decidable reasoning procedures still exist.

On the other hand, nonmonotonic rules have also proven to be a very useful tool for

* Supported by FCT Scholarship SFRH / BD / 38214 / 2007

knowledge representation. They complement the expressive power of DLs, adding the possibility to reason with incomplete information using default negation, and offering natural ways of expressing exceptions, integrity constraints and complex queries. Their formal underpinning lies with declarative, well-understood semantics, the stable model semantics (Gelfond and Lifschitz 1988) and its tractable approximation, the well-founded semantics (Gelder et al. 1991), being the most prominent and widely accepted.

This has led to the need to integrate these distinct knowledge representation paradigms. Over the last decade, there have been many proposals for integrating DLs with nonmonotonic rules (see (Hitzler and Parsia 2009) for a survey). One of the more mature proposals are Hybrid MKNF Knowledge Bases (Motik and Rosati 2007) that allow predicates to be defined concurrently in both an ontology and a set of rules, while enjoying several important properties. A tractable variant of this formalism, based on the well-founded semantics, allows for a top-down querying procedure (Knorr et al. 2011), making the approach amenable to practical applications that need to deal with large knowledge bases.

While such formalisms make it possible to seamlessly combine rules and ontologies in a single unified framework, they do not take into account the highly dynamic character of application areas where they are to be used. In (Slota and Leite 2010b) we made a first step towards a solution to this problem, and decided to address *updates*, a change operation on a knowledge base that records a change that occurred in the modelled world.

Update operators have first been studied in the context of action theories and relational databases with null values (Winslett 1988; Winslett 1990). The basic intuition behind these operators is that the models of a knowledge base represent possible states of the world and when a change in the world needs to be recorded, each of these possible worlds should be modified as little as possible in order to arrive at a representation of the world after the update. This means that, in each possible world, each propositional atom retains its truth value as long as there is no update that directly requires it to change. In other words, *inertia* is applied to the atoms of the underlying language. Later, these operators were successfully applied to partially address updates of DL ontologies (Liu et al. 2006; Giacomo et al. 2006).

But when updates were studied in the context of rules, most authors found atom inertia unsatisfactory. One of the main reasons for this is the clash between atom inertia and the property of *support* (Apt et al. 1988; Dix 1995), which lies at the heart of most logic programming semantics. For instance, when updating a logic program $\mathcal{P} = \{ p \leftarrow q., q. \}$ by $\mathcal{Q} = \{ \sim q. \}$,[1] atom inertia dictates that p must stay true after the update because the update itself does not in any way directly affect the truth value of p. Example 7 in (Giacomo et al. 2006), where a similar update is performed on an analogical DL ontology, shows that such a behaviour may be desirable. However, from a logic programming point of view, one expects p to become false after the update. This is because when q ceases being true, the reason for p to be true disappears as it is no longer supported by any rule.

These intuitions, together with a battery of intuitive examples (Leite and Pereira 1997; Alferes et al. 2000), led to the introduction of the *causal rejection principle* (Leite and Pereira 1997) and subsequently to several approaches to rule updates (Alferes et al. 2000; Eiter et al. 2002; Leite 2003; Alferes et al. 2005) that are fundamentally different from

[1] The symbol $\sim$ denotes default negation.

classical update operators. The basic unit of information to which inertia is applied is no longer an atom, but a rule. This means that a rule stays in effect as long as it does not directly contradict a newer rule. The truth values of atoms are not directly subject to inertia, but are used to determine the set of rules that are overridden by newer rules, and are themselves determined by the remaining rules.

However, the dichotomy between classical and rule updates goes far beyond the different units of information to which inertia is applied. While classical updates are performed on the models of a knowledge base, which renders them syntax-independent, the property of support, being syntactic in its essence, forces rule update methods to refer to the syntactic structure of underlying programs – the individual rules they contain and, in many cases, also the heads and bodies of these rules. As we have shown in (Slota and Leite 2010a), even when classical updates are applied to SE-models (Lifschitz et al. 2001; Turner 2003), a monotonic semantics for logic programs that is more expressive than stable models, the property of support is lost. On the other hand, applying rule updates to DL ontologies leads to a range of technical difficulties. Some of them are caused by the fact that rule update methods are specifically tailored towards identifying and resolving conflicts between pairs of rules. DL axioms do not have a rule-like structure, and a group of pairwise consistent axioms may enter in a conflict. Other difficulties stem from the fact that such a syntactic approach can only hardly offer results similar to those of classical updates, where reasoning by cases is inherent in updating each model of a knowledge base independently of all others. Thus, no single method seems suitable for updating hybrid knowledge bases. A general update operator for hybrid knowledge must somehow integrate the apparently irreconcilable approaches to dealing with evolving knowledge.

In (Slota and Leite 2010b) we simplified this hard task by keeping rules static and allowing the ontology component of a hybrid knowledge base to evolve. Despite the importance of this first step, the applicability of the operator is considerably diminished since typically all parts of a knowledge base are subject to change. As an example, consider the following scenario where both ontologies and rules are needed to assess the risk of imported cargo.

Example 1 (*A Hybrid 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.[2] 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; at a more general level as well. For instance, while the broad HTS code for tomatoes (0702)

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

does not change, the full classification and tariffs for cherry tomatoes for import into the US changes seasonally.

Figure 1 shows a simplified fragment $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$ of such a knowledge base. In this fragment, a shipment has several attributes: the country of its origination, the commodity it contains, its importer, and its producer. The ontology contains a geographic classification, along with information about producers who are located in various countries. It also contains a classification of commodities based on their harmonised tariff information (HTS chapters, headings and codes, cf. `http://www.usitc.gov/tata/hts`). Tariff information is also present, based on the classification of commodities. Finally, the ontology contains (partial) information about three shipments: s_1, s_2 and s_3. There is also a set of rules indicating information about importers, and about whether to inspect a shipment either to check for compliance of tariff information or for food safety issues.

In this paper, we define an update semantics for hybrid knowledge bases that can be used to deal with scenarios such as the one described above. As a theoretical basis for this operator, we first establish a splitting theorem for Hybrid MKNF Knowledge Bases, analogical to the splitting theorem for logic programs (Lifschitz and Turner 1994). The underlying notions then serve us as theoretical ground for identifying a constrained class of hybrid knowledge bases for which a plausible update semantics can be defined by modularly combining a classical and a rule update operator. We then examine basic properties of this semantics, showing that it

- generalises Hybrid MKNF Knowledge Bases (Motik and Rosati 2007).
- generalises the classical minimal change update operator (Winslett 1990).
- generalises the refined dynamic stable model semantics (Alferes et al. 2005).
- adheres to the principle of primacy of new information (Dalal 1988).

Finally, we demonstrate that it properly deals with nontrivial updates in scenarios such as the one described in Example 1.

The rest of this document is structured as follows: We introduce the necessary theoretical background in Sect. 2. Then, in Sect. 3, we establish the splitting theorem for Hybrid MKNF Knowledge Bases, identify a constrained class of such knowledge bases and define a plausible update operator for it. We also take a closer look at its properties and show how it can be applied to deal with updates of the knowledge base introduced in Example 1. We then discuss our results in Sect. 4 and point towards desirable future developments.[3]

2 Preliminaries

In this section we present the formal basis for our investigation. We introduce the unifying semantic framework of Hybrid MKNF Knowledge Bases (Motik and Rosati 2007) that gives a semantics to a knowledge base composed of both DL axioms and rules. Since our hybrid update operator is based on a modular combination a classical and a rule update operator, we introduce a pair of such operators known from the literature and shortly discuss the choices we made.

[3] At `http://centria.di.fct.unl.pt/~jleite/iclp11full.pdf` the reader can find an extended version of this paper with proofs.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\langle p_1, portugal\rangle : \mathsf{RegisteredProducer}$

$portugal : \mathsf{EUCountry}$

$\langle s_1, c_1\rangle : \mathsf{ShpmtCommod}$

$\langle s_1, i_1\rangle : \mathsf{ShpmtImporter}$

$\langle s_2, c_2\rangle : \mathsf{ShpmtCommod}$

$\langle s_2, i_2\rangle : \mathsf{ShpmtImporter}$

$\langle s_2, portugal\rangle : \mathsf{ShpmtCountry}$

$\langle s_3, c_3\rangle : \mathsf{ShpmtCommod}$

$\langle s_3, i_3\rangle : \mathsf{ShpmtImporter}$

$\langle s_3, portugal\rangle : \mathsf{ShpmtCountry}$

$\langle p_2, slovakia\rangle : \mathsf{RegisteredProducer}$

$slovakia : \mathsf{EUCountry}$

$\langle s_1, \text{'07020020'}\rangle : \mathsf{ShpmtDeclHTSCode}$

$c_1 : \mathsf{CherryTomato} \quad c_1 : \mathsf{Bulk}$

$\langle s_2, \text{'07020020'}\rangle : \mathsf{ShpmtDeclHTSCode}$

$c_2 : \mathsf{CherryTomato} \quad c_2 : \mathsf{Prepackaged}$

$\langle s_3, \text{'07020010'}\rangle : \mathsf{ShpmtDeclHTSCode}$

$c_3 : \mathsf{GrapeTomato} \quad c_3 : \mathsf{Bulk}$

$\langle s_3, p_1\rangle : \mathsf{ShpmtProducer}$

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

$\mathsf{CommodCountry}(C, Country) \leftarrow \mathsf{ShpmtCommod}(S, C), \mathsf{ShpmtCountry}(S, Country).$

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

$\mathsf{ExpeditableImporter}(C, I) \leftarrow \mathsf{AdmissibleImporter}(I), \mathsf{ApprovedImporterOf}(I, C).$

$\mathsf{SuspectedBadGuy}(i_1).$

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

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

$\mathsf{CompliantShpmt}(S) \leftarrow \mathsf{ShpmtCommod}(S, C), \mathsf{HTSCode}(C, D), \mathsf{ShpmtDeclHTSCode}(S, D).$

$\mathsf{RandomInspection}(S) \leftarrow \mathsf{ShpmtCommod}(S, C), \mathsf{Random}(C).$

$\mathsf{PartialInspection}(S) \leftarrow \mathsf{RandomInspection}(S).$

$\mathsf{PartialInspection}(S) \leftarrow \mathsf{ShpmtCommod}(S, C), {\sim}\mathsf{LowRiskEUCommodity}(C).$

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

$\mathsf{FullInspection}(S) \leftarrow \mathsf{ShpmtCommod}(S, C), \mathsf{Tomato}(C), \mathsf{ShpmtCountry}(S, slovakia).$

Fig. 1. A Hybrid Knowledge Base for Cargo Imports

MKNF. The logic of Minimal Knowledge and Negation as Failure (MKNF) (Lifschitz 1991) forms the logical basis of Hybrid MKNF Knowledge Bases. It is an extension of first-order logic with two modal operators: $\mathbf{K}$ and $\mathbf{not}$. We use the variant of this logic introduced in (Motik and Rosati 2007). We assume a function-free first-order syntax extended by the mentioned modal operators in a natural way. We denote the set of all predicate symbols by $\mathbf{P}$ and the set of all predicate symbols occurring in a formula ϕ by $\mathsf{pr}(\phi)$.

As in (Motik and Rosati 2007), we only consider Herbrand interpretations in our semantics. We adopt the *standard names assumption*, and apart from the constants used in formulae, we assume our signature to contain a countably infinite supply of constants. The Herbrand Universe of such a signature is denoted by Δ. The set of all (Herbrand) interpretations is denoted by $\mathcal{I}$. An *MKNF structure* is a triple $\langle I, M, N \rangle$ where I is an interpretation and M, N are sets of Herbrand interpretations. The satisfiability of a ground atom p and of an MKNF sentence ϕ in $\langle I, M, N \rangle$ is defined as follows

$$\langle I, M, N \rangle \models p \text{ iff } I \models p$$
$$\langle I, M, N \rangle \models \neg\phi \text{ iff } \langle I, M, N \rangle \not\models \phi$$
$$\langle I, M, N \rangle \models \phi_1 \wedge \phi_2 \text{ iff } \langle I, M, N \rangle \models \phi_1 \text{ and } \langle I, M, N \rangle \models \phi_2$$
$$\langle I, M, N \rangle \models \exists x : \phi \text{ iff } \langle I, M, N \rangle \models \phi[c/x] \text{ for some } c \in \Delta$$
$$\langle I, M, N \rangle \models \mathbf{K}\,\phi \text{ iff } \langle J, M, N \rangle \models \phi \text{ for all } J \in M$$
$$\langle I, M, N \rangle \models \mathbf{not}\,\phi \text{ iff } \langle J, M, N \rangle \not\models \phi \text{ for some } J \in N$$

The symbols $\vee, \forall$ and $\subset$ are interpreted as usual. An *MKNF interpretation* M is a nonempty set of interpretations. Given a set $\mathcal{T}$ of MKNF sentences, we say M is an *S5 model* of $\mathcal{T}$, written $M \models \mathcal{T}$, if $\langle I, M, M \rangle \models \phi$ for every $\phi \in \mathcal{T}$ and all $I \in M$. If there exists the greatest S5 model M of $\mathcal{T}$, then it is denoted by $\mathsf{mod}(\mathcal{T})$. If $\mathcal{T}$ has no S5 model, then $\mathsf{mod}(\mathcal{T})$ denotes the empty set. For all other sets of formulae, $\mathsf{mod}(\mathcal{T})$ stays undefined. M is an *MKNF model* of $\mathcal{T}$ if M is an S5 model of $\mathcal{T}$ and for every MKNF interpretation $M' \supsetneq M$ there is some $I' \in M'$ such that $\langle I', M', M \rangle \not\models \phi$ for some $\phi \in \mathcal{T}$.

Description Logics. Description Logics (Baader et al. 2003) are (usually) decidable fragments of first-order logic that are frequently used for knowledge representation and reasoning in applications. Throughout the paper we assume that some Description Logic is used to describe an ontology, i.e. it is used to specify a shared conceptualisation of a domain of interest. Basic building blocks of such a specification are *constants*, representing objects (or individuals), *concepts*, representing groups of objects, and *roles*, representing binary relations between objects and properties of objects. Typically, an ontology is composed of two distinguishable parts: a TBox specifying the required terminology, i.e. concept and role definitions, and an ABox with assertions about constants.

Most Description Logics can be equivalently translated into function-free first-order logic, with constants represented by constant symbols, atomic concepts represented by unary predicates and atomic roles represented by binary predicates. We assume that for any DL axiom ϕ, $\zeta(\phi)$ denotes such a translation of ϕ. We also define $\mathsf{pr}(\phi)$ as $\mathsf{pr}(\zeta(\phi))$.

Generalised Logic Programs. We consider ground logic programs for specifying non-monotonic domain knowledge. The basic syntactic blocks of such programs are ground atoms. A *default literal* is a ground atom preceded by $\sim$. A *literal* is either a ground atom or a default literal. As a convention, due to the semantics of rule updates that we adopt in what follows, double default negation is absorbed, so that $\sim\sim p$ denotes the atom p. A *rule* r is an expression of the form $L_0 \leftarrow L_1, L_2, \ldots, L_k$ where k is a natural number and $L_0, L_1, \ldots, L_k$ are literals. We say $H(r) = L_0$ is the *head of* r and $B(r) = \{ L_1, L_2, \ldots, L_n \}$ is the *body of* r. A rule r is a *fact* if its body is empty; r is *positive* if its head is an atom. A *generalised logic program* (GLP) $\mathcal{P}$ is a set of rules. The set of predicate symbols occurring in a literal L, set of literals B and a rule r is denoted by $\mathsf{pr}(L)$, $\mathsf{pr}(B)$ and $\mathsf{pr}(r)$, respectively. An interpretation I is a *stable model* of a GLP $\mathcal{P}$ if $I' = \mathit{least}(\mathcal{P} \cup \{ \sim p \mid p \text{ is an atom and } p \notin I \})$, where $I' = I \cup \{ \mathit{not\_p} \mid p \text{ is an atom and } p \notin I \}$ and $\mathit{least}(\cdot)$ denotes the least model of the program obtained from the argument program by replacing every default literal $\sim p$ by a fresh atom $\mathit{not\_p}$.

Hybrid MKNF Knowledge Bases. A hybrid knowledge base is a pair $\langle \mathcal{O}, \mathcal{P} \rangle$ where

$\mathcal{O}$ is an ontology and $\mathcal{P}$ is a generalised logic program. The semantics is assigned to a hybrid knowledge base using a translation function π that translates both ontology axioms and rules into MKNF sentences. For any ontology $\mathcal{O}$, ground atom p, set of literals B, rule r, program $\mathcal{P}$ and hybrid knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$, we define: $\pi(\mathcal{O}) = \{ \mathbf{K}\, \zeta(\phi) \mid \phi \in \mathcal{O} \}$, $\pi(p) = \mathbf{K}\, p$, $\pi({\sim}p) = \mathbf{not}\, p$, $\pi(B) = \{ \pi(L) \mid L \in B \}$, $\pi(r) = (\pi(H(r)) \subset \bigwedge \pi(B(r)))$, $\pi(\mathcal{P}) = \{ \pi(r) \mid r \in \mathcal{P} \}$ and $\pi(\mathcal{K}) = \pi(\mathcal{O}) \cup \pi(\mathcal{P})$. An MKNF interpretation M is an *S5 (MKNF) model of* $\mathcal{K}$ if M is an S5 (MKNF) model of $\pi(\mathcal{K})$.

As was shown in (Lifschitz 1991), the MKNF semantics generalises the stable model semantics for logic programs (Gelfond and Lifschitz 1988) – for every logic program $\mathcal{P}$, the stable models of $\mathcal{P}$ directly correspond to MKNF models of $\pi(\mathcal{P})$.

Classical Updates. As a basis for our update operator, we adopt an update semantics called the *minimal change update semantics* (Winslett 1990) for updating first-order theories. This update semantics offers a simple realisation of atom inertia, satisfies all Katsuno and Mendelzon's postulates for belief update (Katsuno and Mendelzon 1991), and it has successfully been used to deal with ABox updates (Liu et al. 2006; Giacomo et al. 2006).

A notion of closeness between interpretations w.r.t. a fixed interpretation I is used to determine the result of an update. This closeness is based on the set of ground atoms that are interpreted differently than in I. For a predicate symbol P and an interpretation I, we denote the set $\{ p \in I \mid \mathsf{pr}(p) = \{ P \} \}$ by $I^{[P]}$. Given interpretations I, J, J', the *difference in the interpretation of P between I and J*, written $\mathit{diff}(P, I, J)$, is the set $(I^{[P]} \setminus J^{[P]}) \cup (J^{[P]} \setminus I^{[P]})$. We say that J is at least as close to I as J', denoted by $J \leq_I J'$, if for every predicate symbol P it holds that $\mathit{diff}(P, I, J)$ is a subset of $\mathit{diff}(P, I, J')$. We also say that J *is closer to I than* J', denoted by $J <_I J'$, if $J \leq_I J'$ and not $J' \leq_I J$.

The minimal change update semantics then keeps those models of the updating theory that are the closest w.r.t. the relation $\leq_I$ to some model I of the original theory. Given an interpretation I, sets of interpretations M, N, and first-order theories $\mathcal{T}, \mathcal{U}$, we define: $I \oplus N = \{ J \in N \mid \neg(\exists J' \in N)(J' <_I J) \}$, $M \oplus N = \bigcup_{I \in M} (I \oplus N)$, and $\mathsf{mod}(\mathcal{T} \oplus \mathcal{U}) = \mathsf{mod}(\mathcal{T}) \oplus \mathsf{mod}(\mathcal{U})$. If $\mathsf{mod}(\mathcal{T} \oplus \mathcal{U})$ is nonempty, we say it is the *minimal change update model of* $\mathcal{T} \oplus \mathcal{U}$. This notion can be naturally generalised to allow for sequences of updates. Formally, given a finite sequence of first-order theories $\mathcal{U} = \langle \mathcal{U}_i \rangle_{i<n}$, we define $\mathsf{mod}(\mathcal{U}) = (\cdots((\mathsf{mod}(\mathcal{U}_0) \oplus \mathsf{mod}(\mathcal{U}_1)) \oplus \mathsf{mod}(\mathcal{U}_2)) \oplus \cdots) \oplus \mathsf{mod}(\mathcal{U}_{n-1})$. If $\mathsf{mod}(\mathcal{U})$ is nonempty, we say it is the *minimal change update model of* $\mathcal{U}$.

Rule Updates. There exists a variety of different approaches to rule change (Leite and Pereira 1997; Alferes et al. 2000; Eiter et al. 2002; Sakama and Inoue 2003; Alferes et al. 2005; Zhang 2006; Osorio and Cuevas 2007; Delgrande et al. 2007; Delgrande et al. 2008; Slota and Leite 2010a). The more recent, purely semantic approaches (Delgrande et al. 2008; Slota and Leite 2010a) are closely related to classical update operators such as Winslett's operator presented above. However, as indicated in (Slota and Leite 2010a), their main disadvantage is that they violate the property of support that lies at the very heart of semantics of logic programs. Out of the approaches that do respect support, only the rule update semantics presented in (Alferes et al. 2005; Zhang 2006) possess another important property: immunity to tautological and cyclic updates. We henceforth adopt the approach taken in (Alferes et al. 2005) because, unlike in (Zhang 2006), it can be applied to any initial program, can easily be used to perform iterative updates and has a lower computational complexity.

A *dynamic logic program* (DLP) is a finite sequence of GLPs. In order to define the semantics for DLPs, based on *causal rejection of rules*, we define the notion of a conflict between rules as follows: two rules r and r' are *conflicting*, written $r \bowtie r'$, if $H(r) = \sim H(r')$. Given a DLP $\mathcal{P} = \langle \mathcal{P}_i \rangle_{i<n}$ and an interpretation I, we use $\rho(\mathcal{P})$ to denote the multiset of all rules appearing in members of $\mathcal{P}$ and introduce the following notation:

$$Rej(\mathcal{P}, I) = \{\, r \mid (\exists i, j, r')(r \in \mathcal{P}_i \wedge r' \in \mathcal{P}_j \wedge i \leq j \wedge r \bowtie r' \wedge I \models B(r')) \,\} \enspace ,$$
$$Def(\mathcal{P}, I) = \{\, \sim p \mid (\neg \exists r \in \rho(\mathcal{P}))(H(r) = p \wedge I \models B(r)) \,\} \enspace .$$

An interpretation I is a *dynamic stable model* of a DLP $\mathcal{P}$ if $I' = least([\rho(\mathcal{P}) \setminus Rej(\mathcal{P}, I)] \cup Def(\mathcal{P}, I))$, where I' and $least(\cdot)$ are as in the definition of a stable model.

3 Splitting and Updating Hybrid Knowledge Bases

Our general objective is to define an update semantics for finite sequences of hybrid knowledge bases, where each component represents knowledge about a new state of the world.

Definition 2 (*Dynamic Hybrid Knowledge Base*)
A *dynamic hybrid knowledge base* is a finite sequence of hybrid knowledge bases.

In this paper we develop an update operator for a particular class of syntactically constrained hybrid knowledge bases. The purpose of the constraints we impose is to ensure that the ontology and rules can be updated separately from one another, and the results can then be combined to obtain a plausible update semantics for the whole hybrid knowledge base. Formally, the update semantics we introduce generalises and modularly combines a classical and a rule update operator.

In order to identify these constraints, we introduce the splitting theorem for Hybrid MKNF Knowledge Bases in Subsect. 3.1. Based on it, we identify a constrained class of dynamic hybrid knowledge bases and define an update operator for that class in Subsect. 3.2. Finally, in Subsect 3.3 we examine basic properties of the operator and illustrate how it can deal with updates to the hybrid knowledge base from Example 1.

3.1 Splitting Theorem

The splitting theorem for Logic Programs (Lifschitz and Turner 1994) is a generalisation of the notion of program stratification. Given a logic program $\mathcal{P}$, a splitting set for $\mathcal{P}$ is a set of atoms U such that the program can be divided in two subprograms, the bottom and the top of $\mathcal{P}$, such that rules in the bottom only contain atoms from U, and no atom from U occurs in the head of any rule from the top. As a consequence, rules in the top of $\mathcal{P}$ cannot influence the stable models of its bottom. The splitting theorem captures this intuition, guaranteeing that each stable model of $\mathcal{P}$ is a union of a stable model X of the bottom of $\mathcal{P}$ and of a stable model Y of a reduced version of the top of $\mathcal{P}$ where atoms belonging to U are interpreted under X. This can be further generalised to sequences of splitting sets that divide a program $\mathcal{P}$ into a sequence of layers. The splitting sequence theorem then warrants that stable models of $\mathcal{P}$ consist of a union of stable models of each of its layers after appropriate reductions.

In the following we generalise these notions to Hybrid MKNF Knowledge Bases (Motik and Rosati 2007). The first definition establishes the notion of a splitting set in this context.

Definition 3 (*Splitting Set*)
A splitting set for a hybrid knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$ is any set of predicate symbols $U \subseteq \mathbf{P}$ such that

1. For every ontology axiom $\phi \in \mathcal{O}$, if $\mathsf{pr}(\phi) \cap U \neq \emptyset$, then $\mathsf{pr}(\phi) \subseteq U$.
2. For every rule $r \in \mathcal{P}$, if $\mathsf{pr}(H(r)) \cap U \neq \emptyset$, then $\mathsf{pr}(r) \subseteq U$;

The set of ontology axioms $\phi \in \mathcal{O}$ such that $\mathsf{pr}(\phi) \subseteq U$ is called the *bottom of* $\mathcal{O}$ *relative to* U and denoted by $b_U(\mathcal{O})$. The set of rules $r \in \mathcal{P}$ such that $\mathsf{pr}(r) \subseteq U$ is called the *bottom of* $\mathcal{P}$ *relative to* U and denoted by $b_U(\mathcal{P})$. The hybrid knowledge base $b_U(\mathcal{K}) = \langle b_U(\mathcal{O}), b_U(\mathcal{P}) \rangle$ is called *bottom of* $\mathcal{K}$ *relative to* U.

The set $t_U(\mathcal{O}) = \mathcal{O} \setminus b_U(\mathcal{O})$ is the *top of* $\mathcal{O}$ *relative to* U. The set $t_U(\mathcal{P}) = \mathcal{P} \setminus b_U(\mathcal{P})$ is the *top of* $\mathcal{P}$ *relative to* U. The hybrid knowledge base $t_U(\mathcal{K}) = \langle t_U(\mathcal{O}), t_U(\mathcal{P}) \rangle$ is the *top of* $\mathcal{K}$ *relative to* U.

Note that instead of defining a splitting set as a set of atoms, as was done in the case of propositional logic programs, we define it as a set of predicate symbols. By doing this, the set of ground atoms with the same predicate symbol is considered either completely included in a splitting set, or completely excluded from it. While this makes our approach slightly less general than it could be if we considered each ground atom individually, we believe the conceptual simplicity is worth this sacrifice. Also, since all TBox axioms are universally quantified, in many cases we would end up adding or excluding the whole set of ground atoms with the same predicate symbol anyway.

Next, we need to define the reduction that makes it possible to properly transfer information from an MKNF model of the bottom of $\mathcal{K}$, and use it to simplify the top of $\mathcal{K}$.

Definition 4 (*Splitting Set Reduct*)
Let U be a splitting set for a hybrid knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$ and $X \in \mathcal{M}$. The *splitting set reduct of* $\mathcal{K}$ *relative to* U *and* X is a hybrid knowledge base $e_U(\mathcal{K}, X) = \langle t_U(\mathcal{O}), e_U(\mathcal{P}, X) \rangle$, where $e_U(\mathcal{P}, X)$ consists of all rules r' such that there exists a rule $r \in t_U(\mathcal{P})$ with the following properties: $X \models \pi(\{ L \in B(r) \mid \mathsf{pr}(L) \subseteq U \})$, $H(r') = H(r)$, and $B(r') = \{ L \in B(r) \mid \mathsf{pr}(L) \subseteq \mathbf{P} \setminus U \}$.

This leads us to the notion of a solution to $\mathcal{K}$ w.r.t. a splitting set U.

Definition 5 (*Solution w.r.t. a Splitting Set*)
Let U be a splitting set for a hybrid knowledge base $\mathcal{K}$. A *solution to* $\mathcal{K}$ *w.r.t.* U is a pair of MKNF interpretations $\langle X, Y \rangle$ such that X is an MKNF model of $b_U(\mathcal{K})$ and Y is an MKNF model of $e_U(\mathcal{K}, X)$.

The splitting theorem now ensures that solutions to $\mathcal{K}$ w.r.t. any splitting set U are in one to one correspondence with the MKNF models of $\mathcal{K}$.

Theorem 6 (*Splitting Theorem for Hybrid MKNF Knowledge Bases*)
Let U be a splitting set for a hybrid knowledge base $\mathcal{K}$. Then M is an MKNF model of $\mathcal{K}$ if and only if $M = X \cap Y$ for some solution $\langle X, Y \rangle$ to $\mathcal{K}$ w.r.t. U.

This result makes it possible to characterise an MKNF model of a hybrid knowledge base in terms of a pair of MKNF models of two layers inside it, such that, as far as the

MKNF semantics is concerned, the first layer is independent of the second. If instead of a single splitting set we consider a sequence of such sets, we can divide a hybrid knowledge in a sequence of layers, keeping similar properties as in the case of a single splitting set.

Definition 7 (*Splitting Sequence*)
A *splitting sequence* for a hybrid knowledge base $\mathcal{K}$ is a monotone, continuous sequence $\langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for $\mathcal{K}$ such that $\bigcup_{\alpha<\mu} U_\alpha = \mathbf{P}$.

The first layer of $\mathcal{K}$ relative to such a splitting sequence is the part of $\mathcal{K}$ that only contains predicates from U_0. Formally, this is exactly the hybrid knowledge base $b_{U_0}(\mathcal{K})$. Furthermore, for every ordinal $\alpha + 1 < \mu$, the corresponding layer of $\mathcal{K}$ is the part of $\mathcal{K}$ that contains predicates from $U_{\alpha+1} \setminus U_\alpha$, and, in addition, predicate symbols from U_α are allowed to appear in rule bodies. Given our notation this can be written as $t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K}))$. The following definition uses these observations and combines them with suitable reductions to introduce a solution w.r.t. a splitting sequence.

Definition 8 (*Solution w.r.t. a Splitting Sequence*)
Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a hybrid knowledge base $\mathcal{K}$. A *solution to* $\mathcal{K}$ *w.r.t.* U is a sequence $\langle X_\alpha \rangle_{\alpha<\mu}$ of MKNF interpretations such that

1. X_0 is an MKNF model of $b_{U_0}(\mathcal{K})$;
2. For any ordinal α such that $\alpha + 1 < \mu$, $X_{\alpha+1}$ is an MKNF model of
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \textstyle\bigcap_{\eta\leq\alpha} X_\eta\right) \quad ;$$
3. For any limit ordinal $\alpha < \mu$, $X_\alpha = \mathcal{I}$.

The splitting sequence theorem now guarantees a one to one correspondence between MKNF models and solutions w.r.t. a splitting sequence.

Theorem 9 (*Splitting Sequence Theorem for Hybrid MKNF Knowledge Bases*)
Let $\langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for a hybrid knowledge base $\mathcal{K}$. Then M is an MKNF model of $\mathcal{K}$ if and only if $M = \bigcap_{\alpha<\mu} X_\alpha$ for some solution $\langle X_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. $\langle U_\alpha \rangle_{\alpha<\mu}$.

A hybrid knowledge base can be split in a number of different ways. For example, $\emptyset$ and $\mathbf{P}$ are splitting sets for any hybrid knowledge base and sequences such as $\langle \mathbf{P} \rangle$, $\langle \emptyset, \mathbf{P} \rangle$ are splitting sequences for any hybrid knowledge base. The following example shows a more elaborate splitting sequence for the Cargo Import knowledge base.

Example 10 (*Splitting the Cargo Import Knowledge Base*)
Consider the hybrid knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$ presented in Fig. 1. One of the nontrivial splitting sequences for $\mathcal{K}$ is $U = \langle U_0, U_1, U_2, U_3, \mathbf{P} \rangle$, where

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

This splitting sequence splits $\mathcal{K}$ in four layers. The first layer contains all ontological knowledge regarding commodity types as well as information about shipments. The second layer contains rules that use information from the first layer together with internal records to classify importers. 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.

3.2 Update Operator

With the concepts and results related to splitting hybrid knowledge bases from the previous subsection, we are now ready to examine the constraints under which a plausible modular update semantics for a hybrid knowledge base can be defined. Obviously, this is the case with hybrid knowledge bases that contain either only ontology axioms, or only rules. We call such knowledge bases *basic*, and define the dynamic MKNF model for basic dynamic knowledge bases by referring to the classical and rule update semantics defined in Sect. 2.

Definition 11 (*Dynamic MKNF Model of a Basic Dynamic Hybrid Knowledge Base*)
We say a hybrid knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$ is *$\mathcal{O}$-based* if $\mathcal{P}$ contains only positive facts; *$\mathcal{P}$-based* if $\mathcal{O}$ is empty; *basic* if it is either $\mathcal{O}$-based or $\mathcal{P}$-based. A dynamic hybrid knowledge base $\boldsymbol{\mathcal{K}} = \langle \mathcal{K}_i \rangle_{i<n}$ is *$\mathcal{O}$-based* if for all $i < n$, $\mathcal{K}_i$ is $\mathcal{O}$-based; *$\mathcal{P}$-based* if for all $i < n$, $\mathcal{K}_i$ is $\mathcal{P}$-based; *basic* if it is either $\mathcal{O}$-based or $\mathcal{P}$-based.

An MKNF interpretation M is a *dynamic MKNF model* of a basic dynamic hybrid knowledge base $\boldsymbol{\mathcal{K}} = \langle \mathcal{K}_i \rangle_{i<n}$, where $\mathcal{K}_i = \langle \mathcal{O}_i, \mathcal{P}_i \rangle$, if either $\boldsymbol{\mathcal{K}}$ is $\mathcal{O}$-based and M is the minimal change update model of $\langle \zeta(\mathcal{O}_i) \cup \mathcal{P}_i \rangle_{i<n}$, or $\boldsymbol{\mathcal{K}}$ is $\mathcal{P}$-based and $M = \{ J \in \mathcal{I} \mid I \subseteq J \}$ for some dynamic stable model I of $\langle \mathcal{P}_i \rangle_{i<n}$.

As can be seen, our definition is slightly more general than described above, as in the case of $\mathcal{O}$-based knowledge bases it allows the program part to contain positive facts. This amounts to the reasonable assumption that positive facts in a logic program carry the same meaning as the corresponding ground first-order atom. As will be seen in the following, this allows us to extend the class of basic hybrid knowledge bases and define dynamic MKNF models for it. To this end, we utilise the splitting-related concepts from the previous subsection. Their natural generalisation for dynamic hybrid knowledge bases follows.

Definition 12 (*Splitting Set and Splitting Sequence*)
A set of predicate symbols U is a splitting set for a dynamic hybrid knowledge base $\boldsymbol{\mathcal{K}} = \langle \mathcal{K}_i \rangle_{i<n}$ if for all $i < n$, U is a splitting set for $\mathcal{K}_i$.

The dynamic hybrid knowledge base $\langle b_U(\mathcal{K}_i) \rangle_{i<n}$ is called the *bottom of $\boldsymbol{\mathcal{K}}$ relative to U* and denoted by $b_U(\boldsymbol{\mathcal{K}})$. The dynamic hybrid knowledge base $\langle t_U(\mathcal{K}_i) \rangle_{i<n}$ is called the *top of $\boldsymbol{\mathcal{K}}$ relative to U* and denoted by $t_U(\boldsymbol{\mathcal{K}})$. Given some $X \in \mathcal{M}$, the dynamic hybrid knowledge base $\langle e_U(\mathcal{K}_i, X) \rangle_{i<n}$ is called the *splitting set reduct of $\boldsymbol{\mathcal{K}}$ relative to U and X* and denoted by $e_U(\boldsymbol{\mathcal{K}}, X)$.

A sequence of sets of predicate symbols U is a splitting sequence for $\boldsymbol{\mathcal{K}}$ if for all $i < n$, U is a splitting sequence for $\mathcal{K}_i$.

In the static case, given a splitting set U, the splitting set theorem guarantees that an MKNF model M of a hybrid knowledge base $\mathcal{K}$ is an intersection of an MKNF model X of $b_U(\mathcal{K})$ and of an MKNF model Y of $e_U(\mathcal{K}, X)$. In the dynamic case, we can use this correspondence to *define* a dynamic MKNF model. More specifically, we can say that M is a dynamic MKNF model of a dynamic hybrid knowledge base $\mathcal{K}$ if M is an intersection of a dynamic MKNF model X of $b_U(\mathcal{K})$ and of a dynamic MKNF model Y of $e_U(\mathcal{K}, X)$. For the definition to be sound, we need to guarantee that X and Y are defined. In other words, $\mathcal{K}$ has to be such that both $b_U(\mathcal{K})$ and $e_U(\mathcal{K}, X)$ are basic. When we move to the more general case of a splitting sequence $U = \langle U_\alpha \rangle_{\alpha<\mu}$, what we need to ensure is that $b_{U_0}(\mathcal{K})$ is basic and for any ordinal α such that $\alpha + 1 < \mu$, $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K}), \bigcap_{\eta<\alpha} X_\eta)$ is also basic. A class of dynamic hybrid knowledge bases that satisfies these conditions can be defined as follows:

Definition 13 (*Updatable Dynamic Hybrid Knowledge Base*)
Let U be a set of predicate symbols. We say a hybrid knowledge base $\mathcal{K}$ is *$\mathcal{O}$-reducible relative to U* if all rules r from $\mathcal{P}$ are positive and $\mathsf{pr}(B(r)) \subseteq U$; *$\mathcal{P}$-reducible relative to U* if $\mathcal{O}$ is empty; *reducible relative to U* if it is either $\mathcal{O}$-reducible or $\mathcal{P}$-reducible relative to U. A dynamic hybrid knowledge base $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$ is *$\mathcal{O}$-reducible relative to U* if for all $i < n$, $\mathcal{K}_i$ is $\mathcal{O}$-reducible relative to U; *$\mathcal{P}$-reducible relative to U* if for all $i < n$, $\mathcal{K}_i$ is $\mathcal{P}$-reducible relative to U; *reducible relative to U* if it is either $\mathcal{O}$-reducible or $\mathcal{P}$-reducible relative to U.

Let $\mathcal{K}$ be a (dynamic) hybrid knowledge base and $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be a splitting sequence for $\mathcal{K}$. We say U is *update-enabling for $\mathcal{K}$* if $b_{U_0}(\mathcal{K})$ is reducible relative to $\emptyset$ and for any α such that $\alpha + 1 < \mu$, the hybrid knowledge base $t_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K}))$ is reducible relative to U_α. We say $\mathcal{K}$ is *updatable* if some update-enabling splitting sequence for $\mathcal{K}$ exists.

The following proposition now guarantees the property of updatable dynamic hybrid knowledge bases that we discussed above.

Proposition 14 (*Layers of an Updatable Dynamic Hybrid Knowledge Base are Basic*)
Let U be an update-enabling splitting sequence for a dynamic hybrid knowledge base $\mathcal{K}$ and $X \in \mathcal{M}$. Then $b_{U_0}(\mathcal{K})$ is a basic dynamic hybrid knowledge base and for any ordinal α such that $\alpha + 1 < \mu$, $e_{U_\alpha}(b_{U_{\alpha+1}}(\mathcal{K}), X)$ is also a basic dynamic hybrid knowledge base.

This result paves the way to the following definition of a solution to an updatable dynamic hybrid knowledge base together with the notion of a dynamic MKNF model w.r.t. an updatable splitting sequence.

Definition 15 (*Solution to an Updatable Dynamic Hybrid Knowledge Base*)
Let $U = \langle U_\alpha \rangle_{\alpha<\mu}$ be an update-enabling splitting sequence for a dynamic hybrid knowledge base $\mathcal{K}$. A *solution to $\mathcal{K}$ w.r.t. U* is a sequence of MKNF interpretations $\langle X_\alpha \rangle_{\alpha<\mu}$ such that

1. X_0 is a dynamic MKNF model of $b_{U_0}(\mathcal{K})$;
2. For any ordinal α such that $\alpha + 1 < \mu$, $X_{\alpha+1}$ is a dynamic MKNF model of
$$e_{U_\alpha}\left(b_{U_{\alpha+1}}(\mathcal{K}), \bigcap\nolimits_{\eta\leq\alpha} X_\eta\right) \quad ;$$
3. For any limit ordinal $\alpha < \mu$, $X_\alpha = \mathcal{I}$.

We say that an MKNF interpretation M is a *dynamic MKNF model of $\mathcal{K}$ w.r.t. U* if $M = \bigcap_{\alpha<\mu} X_\alpha$ for some solution $\langle X_\alpha \rangle_{\alpha<\mu}$ to $\mathcal{K}$ w.r.t. U.

The last step required to define a dynamic MKNF model of an updatable dynamic hybrid knowledge base, without the need to refer to a context of a particular splitting sequence, is to ensure that Def. 11 of a dynamic MKNF model for basic dynamic hybrid knowledge bases is properly generalised. The following proposition guarantees that the set of dynamic MKNF models is independent of a particular update-enabling splitting sequence.

Proposition 16 (*Solution Independence*)
Let U, V be update-enabling splitting sequences for a dynamic hybrid knowledge base $\mathcal{K}$. Then M is a dynamic MKNF model of $\mathcal{K}$ w.r.t. U if and only if M is a dynamic MKNF model of $\mathcal{K}$ w.r.t. V.

If $\mathcal{K}$ is a basic dynamic hybrid knowledge base, then it can be verified easily that dynamic MKNF models of $\mathcal{K}$, as originally defined in Def. 11, coincide with dynamic MKNF models of $\mathcal{K}$ w.r.t. the splitting sequence $\langle \mathbf{P} \rangle$. We obtain the following corollary:

Corollary 17 (*Compatibility with Def. 11*)
Let $\mathcal{K}$ be a basic dynamic hybrid knowledge base and U be a splitting sequence for $\mathcal{K}$. Then M is a dynamic MKNF model of $\mathcal{K}$ if and only if M is a dynamic MKNF model of $\mathcal{K}$ w.r.t. U.

We can now safely introduce the dynamic MKNF model for any updatable dynamic hybrid knowledge base as follows:

Definition 18 (*Dynamic MKNF Model of Updatable Dynamic Hybrid Knowledge Base*)
An MKNF interpretation M is a *dynamic MKNF model* of an updatable dynamic hybrid knowledge base $\mathcal{K}$ if M is a dynamic MKNF model of $\mathcal{K}$ w.r.t. some update-enabling splitting sequence for $\mathcal{K}$.

3.3 Properties and Use

The purpose of this section is to twofold. First, we establish the most basic properties of the defined update semantics, relating it to the static MKNF semantics and the adopted classical and rule update semantics and showing that it respects one of the most widely accepted principles behind update semantics in general, the principle of primacy of new information. Second, we illustrate its usefulness by considering updates of the hybrid knowledge base presented in Example 1.

The first result shows that our update semantics generalises the static MKNF semantics.

Theorem 19 (*Generalisation of MKNF Models*)
Let $\mathcal{K}$ be an updatable hybrid knowledge base and M be an MKNF interpretation. Then M is a dynamic MKNF model of $\langle \mathcal{K} \rangle$ if and only if M is an MKNF model of $\mathcal{K}$.

It also generalises the classical and rule update semantics it is based on.

Theorem 20 (*Generalisation of Minimal Change Update Semantics*)
Let $\langle \mathcal{K}_i \rangle_{i<n}$, where $\mathcal{K}_i = \langle \mathcal{O}_i, \mathcal{P}_i \rangle$, be a dynamic hybrid knowledge base such that $\mathcal{P}_i$ is empty for all $i < n$. Then M is a dynamic MKNF model of $\langle \mathcal{K}_i \rangle_{i<n}$ if and only if M is the minimal change update model of $\langle \zeta(\mathcal{O}_i) \rangle_{i<n}$.

Theorem 21 (*Generalisation of Dynamic Stable Model Semantics*)
Let $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$, where $\mathcal{K}_i = \langle \mathcal{O}_i, \mathcal{P}_i \rangle$, be a dynamic hybrid knowledge base such that $\mathcal{O}_i$ is empty for all $i < n$. Then M is a dynamic MKNF model of $\mathcal{K}$ if and only if $M = \{ J \in \mathcal{I} \mid I \subseteq J \}$ for some dynamic stable model I of $\langle \mathcal{P}_i \rangle_{i<n}$.

Besides, the semantics respects the principle of primacy of new information (Dalal 1988).

Theorem 22 (*Principle of Primacy of New Information*)
Let $\mathcal{K} = \langle \mathcal{K}_i \rangle_{i<n}$ be an updatable dynamic hybrid knowledge base with $n > 0$ and M be a dynamic MKNF model of $\mathcal{K}$. Then $M \models \pi(\mathcal{K}_{n-1})$.

The following example illustrates how the semantics can be used in the Cargo Imports domain to incorporate new, conflicting information into a hybrid knowledge base.

Example 23 (*Updating the Cargo Import Knowledge Base*)
The hybrid knowledge base $\mathcal{K}$ in Fig. 1 has a single MKNF model M. We shortly summarise what is entailed by this model. First, since the shipments s_1, s_2, s_3 differ in the kind of tomatoes and their packaging, each of them is associated a different tariff charge. The HTS codes of commodities inside all three shipments match the declared HTS codes, so $\mathsf{CompliantShipment}(s_i)$ is entailed for all i. The rules for importers imply that while both $\mathsf{AdmissibleImporter}(i_2)$ and $\mathsf{AdmissibleImporter}(i_3)$ are true, $\mathsf{AdmissibleImporter}(i_1)$ is not true because i_1 is a suspected bad guy. It also follows that $\mathsf{ApprovedImporterOf}(i_2, c_2)$ and $\mathsf{ApprovedImporterOf}(i_3, c_3)$ hold and because of that $\mathsf{ExpeditableImporter}(c_2, i_2)$ and $\mathsf{ExpeditableImporter}(c_3, i_3)$ are also true. Both of these shipments come from a European country, so c_2 and c_3 belong to $\mathsf{LowRiskEUCommodity}$. But this is not true for c_1 since there is no expeditable importer for it. Consequently, $\mathsf{PartialInspection}(s_1)$ holds.

We now consider an update caused by several independent events in order to illustrate different aspects of our hybrid update semantics.

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 rat infestation on the boat with shipment s_3, c_3 is no longer considered a low risk commodity. Finally, due to 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 $\mathcal{K}' = \langle \mathcal{O}', \mathcal{P}' \rangle$: where $\mathcal{O}'$ contains $c_1 : \mathsf{GrapeTomato}$ and $c_3 : (\neg \mathsf{LowRiskEUCommodity})$ as well as all TBox axioms from $\mathcal{O}$,[4] and $\mathcal{P}'$ contains the following rules:[5]

$$\sim\mathsf{ApprovedImporterOf}(i_2, C) \leftarrow \mathsf{Tomato}(C).$$
$$\sim\mathsf{PartialInspection}(S) \leftarrow \mathsf{ShpmtProducer}(S, P), \mathsf{EURegisteredProducer}(P).$$

Note that the splitting sequence U defined in Example 10 is update-enabling for the dynamic hybrid knowledge base $\langle \mathcal{K}, \mathcal{K}' \rangle$. This dynamic hybrid knowledge base has a single dynamic MKNF model M' that entails the following:[6] (1) Commodity c_1 is no longer a member of $\mathsf{CherryTomato}$ and its HTS code is now '07020010'. As a consequence,

[4] We reinclude all TBox axioms in $\mathcal{O}'$ in order to keep them static throughout the example.
[5] We assume that all rule variables are DL-safe and rules are grounded prior to applying our theory.
[6] A more complete explanation including technical details can be found at `http://centria.di.fct.unl.pt/~jleite/iclp11full.pdf`.

$\mathsf{CompliantShpmt}(s_1)$ does not hold, so $\mathsf{FullInspection}(s_1)$ holds. (2): Due to the rule update i_2 is no longer an approved importer for c_2. Thus, c_2 is no longer a member of $\mathsf{LowRiskEUCommodity}$ and $\mathsf{PartialInspection}(s_2)$ holds. (3): A similar situation occurs with c_3 because the update directly asserts that c_3 belongs to the complement of $\mathsf{LowRiskEUCommodity}$. But due to the update of $\mathsf{PartialInspection}$, the fact that c_3 was produced by p_1, and that $\mathsf{EURegisteredProducer}(p_1)$ holds, $\mathsf{PartialInspection}(s_3)$ is not true even though both $\mathsf{ShpmtCommod}(s_3, c_3)$ and $\sim\mathsf{LowRiskEUCommodity}(c_3)$ are true.

4 Discussion

The class of updatable hybrid knowledge bases for which we defined an update semantics in the previous section is closely related to multi-context systems (Brewka and Eiter 2007). Each layer of a hybrid knowledge base relative to a particular update-enabling splitting sequence can be viewed as a context together with all its 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 model semantics, or it contains only DL axioms so that first-order logic can be used as its logic. On the other hand, different splitting sequences induce different multi-context systems, though their overall semantics stays the same. We believe that a further study of this close relationship may bring about new insights.

Another direction in which the proposed framework can be generalised is by letting the ontology and rule update operators be given as parameters instead of using a fixed pair.[7] This seems to have even more appeal given the fact that no general consensus has been reached in the community regarding the "right way" to perform rule updates, and the situation with ontology update operators also seems to be similar. Although Winslett's operator has been used to deal with ABox updates (Liu et al. 2006; Giacomo et al. 2006), its use for dealing with TBox updates has recently been criticised (Calvanese et al. 2010; Slota and Leite 2010b) and a number of considerably different methods for dealing with TBox evolution have been proposed (Qi et al. 2006; Qi and Du 2009; Yang et al. 2009; Calvanese et al. 2010; Wang et al. 2010), many tailored to a specific Description Logic.

To sum up, the contribution of this paper is twofold. First, we generalised the splitting theorems for Logic Programs (Lifschitz and Turner 1994) to the case of Hybrid MKNF Knowledge Bases (Motik and Rosati 2007). This makes it possible to divide a hybrid knowledge base into layers and guarantees that its overall semantics can be reconstructed from the semantics of layers inside it. Second, we used the theorem and related notions to identify a class of hybrid knowledge bases for which we successfully defined an update semantics, based on a modular combination of a classical and a rule update semantics. We showed that our semantics properly generalises the semantics it is based on, particularly the static semantics of Hybrid MKNF Knowledge Bases (Motik and Rosati 2007), the classical minimal change update semantics (Winslett 1990), and the refined dynamic stable model semantics for rule updates (Alferes et al. 2005). We then illustrated on an example motivated by a real world application how the defined semantics deals with nontrivial updates, automatically resolving conflicts and propagating new information across the hybrid knowledge base.

[7] We would like to thank the anonymous reviewer for pointing this out.

References

ALFERES, J. J., BANTI, F., BROGI, A., AND LEITE, J. A. 2005. The refined extension principle for semantics of dynamic logic programming. *Studia Logica 79,* 1, 7–32.

ALFERES, J. J., LEITE, J. A., PEREIRA, L. M., PRZYMUSINSKA, H., AND PRZYMUSINSKI, T. C. 2000. Dynamic updates of non-monotonic knowledge bases. *The Journal of Logic Programming 45,* 1-3 (September/October), 43–70.

APT, K. R., BLAIR, H. A., AND WALKER, A. 1988. Towards a theory of declarative knowledge. In *Foundations of Deductive Databases and Logic Programming.* Morgan Kaufmann, 89–148.

BAADER, F., CALVANESE, D., MCGUINNESS, D. L., NARDI, D., AND PATEL-SCHNEIDER, P. F., Eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

BREWKA, G. AND EITER, T. 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*. AAAI Press, Vancouver, British Columbia, Canada, 385–390.

CALVANESE, D., KHARLAMOV, E., NUTT, W., AND ZHELEZNYAKOV, D. 2010. Evolution of DL-Lite knowledge bases. In *International Semantic Web Conference (1)*, P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, Eds. Lecture Notes in Computer Science, vol. 6496. Springer, Shanghai, China, 112–128.

DALAL, M. 1988. Investigations into a theory of knowledge base revision. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI 1988)*. AAAI Press / The MIT Press, St. Paul, MN, USA, 475–479.

DELGRANDE, J. P., SCHAUB, T., AND TOMPITS, H. 2007. A preference-based framework for updating logic programs. In *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2007)*, C. Baral, G. Brewka, and J. S. Schlipf, Eds. Lecture Notes in Computer Science, vol. 4483. Springer, Tempe, AZ, USA, 71–83.

DELGRANDE, J. P., SCHAUB, T., TOMPITS, H., AND WOLTRAN, S. 2008. Belief revision of logic programs under answer set semantics. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, G. Brewka and J. Lang, Eds. AAAI Press, Sydney, Australia, 411–421.

DIX, J. 1995. A classification theory of semantics of normal logic programs: II. Weak properties. *Fundamenta Informaticae 22,* 3, 257–288.

EITER, T., FINK, M., SABBATINI, G., AND TOMPITS, H. 2002. On properties of update sequences based on causal rejection. *Theory and Practice of Logic Programming (TPLP) 2,* 6, 721–777.

GELDER, A. V., ROSS, K. A., AND SCHLIPF, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM 38,* 3, 620–650.

GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference and Symposium on Logic Programming (ICLP/SLP 1988)*, R. A. Kowalski and K. A. Bowen, Eds. MIT Press, Washington, 1070–1080.

GIACOMO, G. D., LENZERINI, M., POGGI, A., AND ROSATI, R. 2006. 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*. AAAI Press, Boston, Massachusetts, USA.

HITZLER, P. AND PARSIA, B. 2009. Ontologies and rules. In *Handbook on Ontologies*, second ed., S. Staab and R. Studer, Eds. International Handbooks on Information Systems. Springer, 111–132.

KATSUNO, H. AND MENDELZON, A. O. 1991. On the difference between updating a knowledge base and revising it. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, J. F. Allen, R. Fikes, and E. Sandewall, Eds. Morgan Kaufmann Publishers, Cambridge, MA, USA, 387–394.

KNORR, M., ALFERES, J. J., AND HITZLER:, P. 2011. Local closed world reasoning with description logics under the well-founded semantics. *Artificial Intelligence*. Accepted for publication.

LEITE, J. A. 2003. *Evolving Knowledge Bases*. Frontiers of Artificial Intelligence and Applications, xviii + 307 p. Hardcover, vol. 81. IOS Press.

LEITE, J. A. AND PEREIRA, L. M. 1997. Generalizing updates: From models to programs. In *Proceedings of the 3rd International Workshop on Logic Programming and Knowledge Representation (LPKR '97)*, J. Dix, L. M. Pereira, and T. C. Przymusinski, Eds. Lecture Notes in Computer Science, vol. 1471. Springer, Port Jefferson, New York, USA, 224–246.

LIFSCHITZ, V. 1991. Nonmonotonic databases and epistemic queries. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI91)*. 381–386.

LIFSCHITZ, V., PEARCE, D., AND VALVERDE, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL) 2,* 4, 526–541.

LIFSCHITZ, V. AND TURNER, H. 1994. Splitting a logic program. In *Proceedings of the 11th International Conference on Logic Programming (ICLP 1994)*, P. V. Hentenryck, Ed. MIT Press, Santa Margherita Ligure, Italy, 23–37.

LIU, H., LUTZ, C., MILIČIĆ, M., AND WOLTER, F. 2006. Updating description logic ABoxes. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, P. Doherty, J. Mylopoulos, and C. A. Welty, Eds. AAAI Press, Lake District of the United Kingdom, 46–56.

MOTIK, B. AND ROSATI, R. 2007. A faithful integration of description logics with logic programming. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, M. M. Veloso, Ed. Hyderabad, India, 477–482.

OSORIO, M. AND CUEVAS, V. 2007. Updates in answer set programming: An approach based on basic structural properties. *Theory and Practice of Logic Programming 7,* 4, 451–479.

QI, G. AND DU, J. 2009. Model-based revision operators for terminologies in description logics. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena (IJCAI 2009)*, C. Boutilier, Ed. California, USA, 891–897.

QI, G., LIU, W., AND BELL, D. A. 2006. A revision-based approach to handling inconsistency in description logics. *Journal of Artificial Intelligence Review 26,* 1-2, 115–128.

SAKAMA, C. AND INOUE, K. 2003. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming (TPLP) 3,* 6, 671713.

SLOTA, M. AND LEITE, J. 2010a. On semantic update operators for answer-set programs. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, H. Coelho, R. Studer, and M. Wooldridge, Eds. Frontiers in Artificial Intelligence and Applications, vol. 215. IOS Press, Lisbon, Portugal, 957–962.

SLOTA, M. AND LEITE, J. 2010b. Towards Closed World Reasoning in Dynamic Open Worlds. *Theory and Practice of Logic Programming, 26th Int'l. Conference on Logic Programming (ICLP'10) Special Issue 10,* 4-6 (July), 547–564.

TURNER, H. 2003. Strong equivalence made easy: nested expressions and weight constraints. *Theory and Practice of Logic Programming (TPLP) 3,* 4-5, 609–622.

WANG, Z., WANG, K., AND TOPOR, R. W. 2010. A new approach to knowledge base revision in DL-Lite. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*, M. Fox and D. Poole, Eds. AAAI Press, Atlanta, Georgia, USA.

WINSLETT, M. 1988. Reasoning about action using a possible models approach. In *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI 1988)*. AAAI Press / The MIT Press, Saint Paul, MN, USA, 89–93.

WINSLETT, M. 1990. *Updating Logical Databases*. Cambridge University Press, New York, USA.

YANG, F., QI, G., AND HUANG, Z. 2009. A distance-based operator to revising ontologies in DL $\mathcal{SHOQ}$. In *Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009)*, C. Sossai and G. Chemello, Eds. Lecture Notes in Computer Science, vol. 5590. Springer, Verona, Italy, 434–445.

ZHANG, Y. 2006. Logic program-based updates. *ACM Transactions on Computational Logic 7,* 3, 421–472.

Appendix A Responses to Reviewers' Comments

We would like to thank each reviewer for the helpful reviews.

We tried to carefully take all of them in consideration while preparing the revised version of our paper. Many suggestions helped clarify the motivation for choices we made in the paper. Unfortunately, due to the strict space limitations, some of the revisions are not as detailed as we would have liked, and some could not be implemented at all. In the following sections, we address the reviewers' specific comments, indicating when they lead to revisions in the paper.

At `http://centria.di.fct.unl.pt/~jleite/iclp11full.pdf`, the reader can find the extended version of this paper including proofs.

A.1 Review 1

> The papers introduces an update operator for a large class of hybrid knowledge bases where both the DL component as well as the rule component are allowed to dynamically change. The paper also introduces splitting sequences and splitting theorem for hybrid knowledge bases, use them to define a modular update semantics, investigate its basic properties, and illustrate its use on a realistic example about cargo imports.
> The paper is interesting but I think it needs more motivation that supports the proposed approach. In fact my main concern is about section 2 of the paper.
> Concrete comments.
> In page 3 the authors wrote:
> "While classical updates are performed on the models of a knowledge base, ... , forces rule update methods to refer to the syntactic structure of underlying programs."
> I think this is true in many but not all the approaches, (see 1,2,3).

As the excluded part of the cited sentence indicates, purely semantic approaches such as 1) and (Slota and Leite 2010a) have serious problems with the property of support that is of great import with respect to logic programs and their established semantics. On the other hand, the work in 2) and 3) falls into the category of semantics that need to refer to the syntactic structure of a program (the individual rules and their heads and bodies) in order to perform an update.

We added a further explanation of what we meant by "syntactic structure" in that sentence, hoping to avoid any further ambiguity.

> The paper presentes in page 5, 4 properties that hold in their approach. Why these properties are important?

We added a paragraph at the beginning of that subsection to better motivate the investigation of these properties.

> I think that it could be more interestingg if you present and discuss properties related to updates as the last property commented in page 5.

Although a discussion of other desirable properties of update operators would be appropriate, it would also require a considerable amount of space. Our main intention in this paper was to show that the substantially different update methods can be integrated in a unifying framework and that this framework is useful in a realistic scenario. Given the strict space limitations, we opted not to include discussion of other properties of update operators.

Note also that the properties we examine prior to the principle of primacy of new information are not good candidates for removal. First, they form important bridges between the framework and the ingredients it was made from. Second, they would serve as important reference points if a further discussion of update properties were to be included. This is because Theorems 20 and 21 implicitly show that the resulting framework cannot satisfy a property that is not satisfied by one of the adopted update semantics. This means, for instance, that due to the results of (Eiter et al. 2002), the framework cannot satisfy most of Katsuno and Mendelzon's postulates for belief update. On the other hand, properties such as Initialization, Idempotence and Absorption (see (Eiter et al. 2002), page 21) do hold.

> The paper consider 3 symbols for negation, please explain such symbols early in the paper.

Due to the integrative nature of the proposed framework, these symbols are presented as part of introduction to the components that are being integrated. While the symbol $\sim$ denotes the usual symbol of default negation in a logic program, the symbols $\neg$ and $\mathbf{not}$ are due to the logic of Minimal Knowledge and Negation as Failure (Lifschitz 1991; Motik and Rosati 2007). We added a footnote with an explanation of the symbol $\sim$ which is the only one that was used before it was introduced.

> The paper says "As a convention, double default negation is absorbed .." Please explain briefly the reason of such convention since in many non-monotonic formalisms, double negation of X is no equivalent to X.

We added an explanation for this convention.

> The paper says in page 7 that "This update semantics offers a simple and intuitive realisation ...". I would prefer to avoid using terms such as "simple and intuitive" without including a real discussion on this issue.

We removed the classification "intuitive" from the sentence.

> Again in page 7 the paper says:
> "we adopt the rule update semantics introduced in (Alferes et al. 2005). The main reason is that, unlike others, it enjoys a combination of important properties, such as immunity to cyclic and tautological updates, and respect for the property of support."
> Here I think thae two first properties are more or less well accepted, but is that enough? As one can see in the definition of page 8 such approach is restricted in two ways. It considers only normal rules and it does not constructs an updated program. Again, see (1,2,3) for other options.

We added a more detailed discussion of rule update semantics to better justify our choice.

> It could be interesting if you could suggest an approach where you consider the ”update operator” as a given parameter. Would that be possible?

We currently rely on the specific properties of the chosen update operators to establish the properties of the overall framework. Abstracting from the specific operators and finding a minimal set of necessary conditions that will guarantee the overall properties is indeed an issue worth pursuing, which we now discuss in Sect. 4.

> I think that the example about cargo imports is useful and interesting. Finally the paper do not present proofs or sketch of the proofs but I as far as I could check I think that the paper is sound.
>
> ---
>
> 1)A General Approach to Belief Change in Answer Set Programing James del Grande, Torsten Schaub, ... http://arxiv.org/abs/0912.5511
> 2) Mauricio Osorio, Víctor Cuevas: Updates in answer set programming: An approach based on basic structural properties. TPLP 7(4):451-479 (2007)
> 3) J. C. Acosta Guadarrama, Jürgen Dix, Mauricio Osorio Galindo and Fernando Zacaras. Updates in Answer Set Programming based on structural properties. In S. McIlraith, P. Peppas, and M. Thielscher, editors, the 7th International Symposium on Logical Formalizations of Commonsense Reasoning, pages 213219, Corfu, Greece, May 2005. `http://www2.in.tu-clausthal.de/~guadarrama/public/`

A.2 Review 2

> This paper presents a generalization of the splitting theorems for Logic Programsto the case of Hybrid MKNF (for Minimal Knowledge and Negation as Failure) Knowledge Bases making it possible to produce a stratification of hybrid knowledge bases guaranteeing that the overall semantics can be reconstructed from the semantics of their strata. The results can be used to identify a class of hybrid knowledge bases for which an update semantics based on a modular combination of a classical and a rule update semantics has been defined. this semantics properly generalizes the static semantics of Hybrid MKNF Knowledge Bases, the classical minimal change update semantics, and the dynamic stable model semantics for rule updates. An application example is presented showing how the defined semantics handles nontrivial updates, solving conflicts and propagating new information through the hybrid knowledge base.
> This is a very interesting paper; it is clearly presented and motivated. The problem it addresses is an important one with many potential applications. While the example is clearly a small one, it shows the possibilities of the approach. In my opinion can be accepted.

A.3 Review 3

> This paper is about splitting and updating MKNF hybrid KBs. The topic is relevant to the conference. The technical results appear to be sound, but much of them are as expected, as they are adapted from the original formulation with no significant difference. The splitting theorem is adapted from the the original theorem by Lifschitz and Turner. True, the splitting theorem by Lifschitz and Turner applies to logic programs, and this work is about hybrid MKNF, but given the close relationship between MKNF and logic programs, the results are as expected. In fact, it becomes simpler than splitting the general MKNF theories, because in hybrid MKNF KB, the syntax of "P" is restricted to a form of logic programs. My guess is that the proof is not too different from the original proof. So it seems to me that the contribution of this paper is to merge the existing technical results (splitting theorem, minimal change update, dynamic stable models) into the context of MKNF hybrid KBs, but the authors should clearly state what contributions they make to MKNF hybrid KB: There are many update semantics for LP and classical logic. Why is the one that the authors adopted relevant, in comparison with alternatives?

In Sect. 2 we added a more detailed discussion to justify our choice of update operators. We also believe that the framework can be further generalised to allow for other than the update operators we have chosen. This is now discussed in Sect. 4.

> In several places intersection is consistently used in place of union. For instance, Theorem 6, Definition 8, Theorem 9, to name a few. Please check.

This is due to the fact that in MKNF the semantic structure is not a single first-order interpretation, but a set of such interpretations. As a consequence, the set-theoretic operations are "reversed". For instance, in Theorem 6, one needs to "join" the information contained in X and in Y. This is done by taking their intersection to constrain the "possible worlds" to only those contained in both X and Y. A similar situation occurs in other places throughout the paper.

> There is only one big example. It will be better if the authors provide small examples to illustrate various definitions and theorems.

Although we agree that the paper would benefit from a number of smaller examples to facilitate understanding the concepts we introduce, we prefer to keep the big example because it shows that the integrated approach is indeed useful when used in a realistic scenario. Due to the strict space limitations, we could not include smaller examples for illustrational purposes.