TLP **10** (4-6): 547–564, July. © 2010 Cambridge University Press
DOI: 10.1017/xxxx Printed in the United Kingdom

Towards Closed World Reasoning in Dynamic Open Worlds

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

submitted 7 February 2010; revised 10 April 2010; accepted 21 May 2010

Abstract

The need for integration of ontologies with nonmonotonic rules has been gaining importance in a number of areas, such as the Semantic Web. A number of researchers addressed this problem by proposing a unified semantics for *hybrid knowledge bases* composed of both an ontology (expressed in a fragment of first-order logic) and nonmonotonic rules. These semantics have matured over the years, but only provide solutions for the static case when knowledge does not need to evolve.

In this paper we take a first step towards addressing the dynamics of hybrid knowledge bases. We focus on knowledge updates and, considering the state of the art of belief update, ontology update and rule update, we show that current solutions are only partial and difficult to combine. Then we extend the existing work on ABox updates with rules, provide a semantics for such evolving hybrid knowledge bases and study its basic properties.

To the best of our knowledge, this is the first time that an update operator is proposed for hybrid knowledge bases.

KEYWORDS: belief change, belief update, hybrid knowledge bases, ontologies, rules, description logics, answer set programming, semantic web

1 Introduction

In this paper we address updates of hybrid knowledge bases composed of a Description Logic ontology and Logic Programming rules. We propose an operator to be used when a hybrid theory is updated by new observations of a changing world, examine its properties, and discuss open problems pointing to future research.

The Semantic Web was initiated almost a decade ago with an ambitious plan regarding the sharing of metadata and knowledge in the Web, enhanced with reasoning services for advanced new applications (Berners-Lee et al. 2001). Since then, the considerable amount of research devoted to this endeavour originated important foundational results and a deeper understanding of the issues involved, while identifying important conclusions regarding future developments, namely that:

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

1. Ontologies are necessary and useful for knowledge representation in the Semantic Web. The formalisms developed, e.g. OWL, are powerful enough to capture existing modelling languages used in software engineering, and extend their capabilities. Ontologies are usually based on decidable, as well as tractable, fragments of Classical Logic, such as the Description Logics (DL) (Baader et al. 2003). They adopt the *open world assumption* (OWA) i.e. they view a knowledge base, *by assumption*, to be potentially incomplete, hence a proposition p is false only if the knowledge base is inconsistent with p. This suits well the open nature of such systems where complete knowledge about the environment cannot be assumed.
2. Rules are fundamental to overcome the limitations found in OWL. They enjoy formal, declarative and well-understood semantics, the *stable model semantics* (Gelfond and Lifschitz 1988) and its tractable approximation, the three-valued *well-founded semantics* (Gelder et al. 1991) being the most prominent and widely accepted. These semantics adopt the *closed world assumption* (CWA) i.e. the knowledge base *is assumed* to contain complete information. Consequently, a proposition p is considered false whenever it is not entailed to be true. This type of negation is usually dubbed *default negation* or *weak negation*, to distinguish it from the *classical negation* used in Classical Logic. Rules can naturally express assumptions, policies, preferences, norms and laws, and provide constructs which are more natural for software developers (as used in Relational Databases and Logic Programming).
3. The open and dynamic character of the Semantic Web requires new knowledge based systems to be equipped with mechanisms to evolve.

Indeed, the growing availability of information requires the support of dynamic data and application integration, automation and interoperation of business processes and problem-solving in various domains, to enforce correctness of decisions, and to allow traceability of the knowledge used and of the decisions taken. In these scenarios, ontologies provide the logical foundation of intelligent access and information integration, while rules are used to represent business policies, regulations and declarative guidelines, and mappings between different information sources.

Over the last decade, there have been many proposals for integrating DL based monotonic ontologies with nonmonotonic rules (see (Hitzler and Parsia 2009) for a survey). Recently, in (Motik and Rosati 2007), Hybrid MKNF Knowledge Bases were introduced, allowing predicates to be defined concurrently in both an ontology and a set of rules, while enjoying several important properties. There is even a tractable variant based on the well-founded semantics that allows for a top-down querying procedure (Alferes et al. 2009), making the approach amenable to practical applications that need to deal with large ontologies.

But this only addresses part of the problem. The highly dynamic character of the Semantic Web calls for the development of ways to deal with updates of these hybrid knowledge bases composed of both rules and ontologies, and the inconsistencies that may arise. The dynamics of hybrid knowledge bases, to the best of our knowledge, has never been addressed before.

However, the problems associated with knowledge evolution have been extensively studied, over the years, by researchers in different research communities, namely in the context of Classical Logic, and in the context of Logic Programming. They proved to be extremely difficult to solve, and existing solutions, even within each community, are still subject of active debate as they do not seem adequate in all kinds of situations in which their application is desirable.

In the context of Classical Logic, the seminal work by Alchourrón, Gärdenfors and Makinson (AGM) (Alchourrón et al. 1985) proposed a set of desirable properties of belief change operators, now called *AGM postulates*. Subsequently, in (Katsuno and Mendelzon 1991), *update* and *revision* have been distinguished as two very related but ultimately different belief change operations. While revision deals with incorporating new information about a static world, update takes place when changes occurring in a dynamic world are recorded. The authors of (Katsuno and Mendelzon 1991) formulated a separate set of postulates for updates. One of the specific update operators that satisfies these postulates is Winslett's minimal change update operator (Winslett 1990). Though we believe that revision operators for hybrid knowledge bases pose an interesting and important research topic, in this paper we focus on update operators and do not tackle revision any further.

Further research showed that, in most cases, belief update operators cannot be directly applied to Description Logic ontologies. The existing work considers only ABox updates, allowing only for static acyclic TBoxes which are "expanded" before the update takes place (Liu et al. 2006), or static general TBoxes in the form of integrity constraints (Giacomo et al. 2007). The main reasons for these restrictions were expressibility and computability of the updated ontology. But we believe there is a more fundamental problem with using belief update operators to update TBoxes because it frequently yields counterintuitive results, as illustrated here:

Example 1 (*Counterintuitive TBox update*)
Suppose we want to update the description logic TBox $\mathcal{T} = \{ B \sqsubseteq A \}$ with the new information $\mathcal{U} = \{ C \sqsubseteq B \}$. In other words, we introduce a new subconcept C of concept B. Using Winslett's update operator we obtain the updated knowledge base $\{C \sqsubseteq B, B \sqcap \neg C \sqsubseteq A\}$. Thus, the subconcept axiom from $\mathcal{T}$ is severely weakened. Using other operators (see (Herzig and Rifi 1999) for a survey) it may even get completely forgotten. However, new subconcepts may arise in the modelled environment without disturbing other relations the target concept may have.

Thus, appropriate ways of updating ontologies in general, and TBoxes in particular, still need to be explored and pose an important open problem on its own. In our current paper we follow the mentioned ontology update literature and focus on ABox updates, leaving the TBox static throughout the update process.

Updates were also investigated in the context of Logic Programs. Earlier approaches based on literal inertia (Marek and Truszczynski 1998) proved not sufficiently expressive for dealing with rule updates, leading to the development of rule update semantics based on different intuitions, principles and constructions, when compared to their classical counterparts. For example, the introduction of the *causal rejection principle* (Leite and Pereira 1997) lead to several approaches to rule updates (Alferes et al. 2000; Leite 2003; Eiter et al. 2002; Alferes et al. 2005), all of them with a strong syntactic flavour which makes them very hard to combine with belief update operators that are semantic in their nature. Other existing approaches to updates of Logic Programs (Sakama and Inoue 2003; Zhang and

Foo 2005; Delgrande et al. 2008) have different problems, such as, for example, not being immune to tautological updates. It has been shown in (Eiter et al. 2002) that the above mentioned rationality postulates, set forth in the context of Classical Logic, are inappropriate for dealing with updates of Logic Programs.

In order to develop an appropriate update operator for hybrid knowledge bases, one has to somehow combine these apparently irreconcilable approaches to updates, a problem that is far away from having an appropriate solution.

In this paper, we take an important first step in addressing the updates of hybrid knowledge bases. Following the state of the art in ontology updates (Liu et al. 2006; Giacomo et al. 2007), we choose a constrained scenario – which is, nevertheless, rich enough to encompass many practical applications of hybrid theories – in which only the ABox is allowed to evolve, while the TBox is kept static. We add rule support to this scenario by augmenting the traditional immediate consequence operator used in logic programming with the classical update operator. The resulting framework is significantly more expressive than those of (Liu et al. 2006; Giacomo et al. 2007) and allows for a seamless two-way interaction between Logic Programming rules and Description Logic axioms. The consequences of rules are also subject to update through the ABox updates, making it possible to use rules to represent default preferences or behaviour and later directly impose exceptions to those rules.

The resulting update semantics enjoys several desirable properties, namely it:

- generalises the stable model semantics (Gelfond and Lifschitz 1988).
- generalises, under reasonable assumptions, the MKNF semantics for hybrid knowledge bases (Motik and Rosati 2007).
- generalises, under reasonable assumptions, the minimal change update operator (Winslett 1990).
- adheres to the principle of primacy of new information (Dalal 1988), so every model resulting from the update by an ABox $\mathcal{A}$ is a model of $\mathcal{A}$.
- is syntax-independent w.r.t. the TBox and ABox, i.e. yields the same result with equivalent TBoxes and when updating by equivalent ABoxes.

To the best of our knowledge, this is the first proposal of an update semantics for hybrid knowledge bases in a single framework. This semantics not only provides an appropriate solution to the constrained scenario we chose, but it unveils a set of important issues, opening the door for interesting future research endeavours.

The remainder of this paper is structured as follows: In Sect. 2 we introduce the notions needed throughout the rest of the paper, and discuss some of the choices we made. Section 3 contains the definition of our operator while in Sect. 4 we examine its properties. In Sect. 5 we conclude and sketch some directions for future work[1].

2 Preliminaries

In this section we present the necessary preliminaries that we need to define the hybrid update operator, and discuss some of the choices we made. As the basis for the formal

[1] At `http://arxiv.org/abs/1004.4342` the reader can find an extended version of this paper with all the proofs.

part of our investigation, we choose the same notation and notions as those used for Hybrid MKNF Knowledge Bases (Motik and Rosati 2007). This makes it possible to treat first-order formulae and nonmonotonic rules in a unified manner and also compare our semantics to the one of Hybrid MKNF more easily.

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 $\mathbf{not}$. In the following, we follow the presentation of syntax and semantics of this logic as given in (Motik and Rosati 2007). We assume a function-free first-order syntax extended by the mentioned modal operators in a natural way. A *first-order atom* is a formula $P(t_1, t_2, \ldots, t_n)$ where P is a predicate symbol of arity n and t_i are terms. An MKNF formula of the form $\mathbf{K}\,\phi$ is called a *modal* $\mathbf{K}$*-atom*, and a formula of the form $\mathbf{not}\,\phi$ is called a *modal* $\mathbf{not}$*-atom*; collectively, modal $\mathbf{K}$- and $\mathbf{not}$-atoms are called *modal atoms*. An MKNF formula ϕ is a *sentence* if it has no free variables; ϕ is *ground* if it does not contain variables; ϕ is *first-order* if it does not contain modal operators. By $\phi[t_1/x_1, t_2/x_2, \ldots, t_n/x_n]$ we denote the formula obtained by simultaneously replacing in ϕ all free occurrences of variable x_i by term t_i. A set of first-order sentences is a *first-order theory*.

Similarly as in (Motik and Rosati 2007), we only consider Herbrand interpretations in our semantics. We adopt the *standard names assumption*, so apart from the constants used in formulae, we assume our signature to contain a countably infinite supply of constants not occurring in the formulae. The Herbrand Universe of such a signature is denoted by Δ and the set of all Herbrand first-order interpretations is denoted by $\mathcal{I}$. An *MKNF structure* is a triple $\langle I, M, N \rangle$ where I is a Herbrand first-order interpretation and M, N are nonempty sets of Herbrand first-order interpretations. The satisfiability of an MKNF sentence ϕ in $\langle I, M, N \rangle$ is defined as follows (where p is a ground first-order atom):

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

The symbols $\vee$, $\forall$ and $\subset$ (material implication) are interpreted as usual. An *MKNF interpretation* M is a nonempty set of Herbrand first-order interpretations over Δ. Let $\mathcal{S}$ be a set of MKNF sentences and M an MKNF interpretation. M is an *S5 model* of $\mathcal{S}$, written $M \models \mathcal{S}$, if $\langle I, M, M \rangle \models \phi$ for every $\phi \in \mathcal{S}$ and all $I \in M$. If there exists the greatest S5 model M of $\mathcal{S}$, then it is denoted by $\mathsf{mod}(\mathcal{S})$. If $\mathcal{S}$ has no S5 model, then $\mathsf{mod}(\mathcal{S})$ denotes the empty set. For all other sets of formulae, $\mathsf{mod}(\mathcal{S})$ stays undefined. M is an *MKNF model* of $\mathcal{S}$ if M is an S5 model of $\mathcal{S}$ 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 $\phi \in \mathcal{S}$. For a sentence ϕ, the S5 models of ϕ, MKNF models of ϕ and $\mathsf{mod}(\phi)$ are defined as S5 models of $\{\,\phi\,\}$, MKNF models of $\{\,\phi\,\}$ and $\mathsf{mod}(\{\,\phi\,\})$.

Description Logics. Description Logics (DLs) (Baader et al. 2003) are (mostly) decidable fragments of first-order logic that are frequently used for knowledge representation in practical applications. In the following we assume that some Description Logic is used to describe an ontology. We do not choose any specific Description Logic, we only assume that the ontology expressed in it is composed of two distinguishable parts: a TBox with concept and role definitions using the constructs of the underlying description logic, and an ABox with individual assertions, i.e. assertions of the form $C(a)$ and $R(a, b)$ where a, b are constants, C is a concept expression and R is a role expression of the underlying description logic. This distinction is important to us as we treat the two types of knowledge in different ways – the TBox is considered static while the ABox is allowed to evolve. As was noted in the introduction, our main reason for this is that we believe existing update operators to be unsuitable for updating concept definitions contained in the TBox. We also assume that the axioms of the underlying DL can be translated into first-order logic and for the sake of simplicity we assume that the TBox and ABox already contain these translations instead of the syntactic constructs of the underlying DL.

Hybrid MKNF Knowledge Bases. We make use of the general MKNF framework to give a semantics to hybrid knowledge bases composed of an ontology and a normal logic program. We define a *rule* to be any formula of the following form

$$\mathbf{K}\, p \subset \mathbf{K}\, q_1 \wedge \mathbf{K}\, q_2 \wedge \cdots \wedge \mathbf{K}\, q_k \wedge \mathbf{not}\, s_1 \wedge \mathbf{not}\, s_2 \wedge \cdots \wedge \mathbf{not}\, s_l \qquad (1)$$

where k, l are non-negative integers and p, q_i, s_j are first-order atoms for any $i \in \{1, 2, \ldots, k\}$, $j \in \{ 1, 2, \ldots, l \}$. Given a rule r of the form (1), the following notation is also defined: $H(r) = \mathbf{K}\, p$, $H^*(r) = p$, $B^+(r) = \{ \mathbf{K}\, q_1, \mathbf{K}\, q_2, \ldots, \mathbf{K}\, q_k \}$, $B^-(r) = \{\mathbf{not}\, s_1, \mathbf{not}\, s_2, \ldots, \mathbf{not}\, s_l\}$ and $B(r) = B^+(r) \cup B^-(r)$. $H(r)$ is dubbed the *head of* r, $H^*(r)$ the *first-order head of* r, $B^+(r)$ the *positive body of* r, $B^-(r)$ the *negative body of* r and $B(r)$ the *body of* r. An MKNF rule r is called *definite* if its negative body is empty; it is called a *fact* if its body is empty. A *program* is a set of rules and a *definite program* is a set of definite rules.

As was shown in (Lifschitz 1991), the MKNF semantics generalises the stable model semantics for logic programs – every logic programming rule of the form

$$p \leftarrow q_1, q_2, \ldots, q_k, not\ s_1, not\ s_2, \ldots, not\ s_l.$$

can be translated into the MKNF formula (1) and the stable models of sets of such rules (i.e. of normal logic programs) directly correspond to MKNF models of the set of translated rules.

We are now ready to define a hybrid knowledge base and its semantics.

Definition 2 (*Hybrid knowledge base*)
Let $\mathcal{O}$ be an ontology and $\mathcal{P}$ a program. The pair $\langle \mathcal{O}, \mathcal{P} \rangle$ is then called a *hybrid knowledge base*.

For an ontology $\mathcal{O}$, a rule r with the vector of free variables $\mathbf{x}$, a program $\mathcal{P}$ and the hybrid knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$, we define: $\pi(\mathcal{O}) = \{ \mathbf{K}\, \phi \mid \phi \in \mathcal{O} \}$, $\pi(r) = (\forall \mathbf{x} : r)$, $\pi(\mathcal{P}) = \{ \pi(r) \mid r \in \mathcal{P} \}$ and $\pi(\mathcal{K}) = \pi(\mathcal{O}) \cup \pi(\mathcal{P})$.

We say an MKNF interpretation M is an *S5 model of* $\mathcal{K}$ if M is an S5 model of $\pi(\mathcal{K})$. We say M is an *MKNF model of* $\mathcal{K}$ if M is an MKNF model of $\pi(\mathcal{K})$.

In this paper, we are not concerned with decidability of reasoning, so we refrain from introducing a safety condition on our rules as was done in (Motik and Rosati 2007).

Classical Updates. As a basis for our update operator, we adopt an update semantics called the *minimal change update semantics* (sometimes also called the *possible models approach* (PMA)) as defined in (Winslett 1990) for updating first-order theories. There are a number of reasons for this choice. First, it satisfies all of Katsuno and Mendelzon's update postulates (Katsuno and Mendelzon 1991). This means, for instance, that unlike some other update semantics, such as the standard semantics (Winslett 1990), it is not sensitive to syntax of the original theory or of the update. Second, it is based on an intuitive idea, treating each classical model of the original theory as a possible world and modifying it as little as possible in order to become consistent with the new information. This idea has its roots in reasoning about action (Winslett 1988) and updates of relational theories (Winslett 1990). Third, the operator has already been successfully used to deal with ABox updates (Liu et al. 2006; Giacomo et al. 2007).

This semantics uses a notion of closeness of first-order interpretations w.r.t. a fixed first-order interpretation I. This notion is based on the set of ground first-order atoms that are interpreted differently than in I.

Definition 3 (*Interpretation distance*)
Let P be a predicate symbol and I, J be first-order interpretations. The *difference in the interpretation of P between I and J*, written $\mathit{diff}(P, I, J)$, is a relation containing the set of tuples $(P^I \setminus P^J) \cup (P^J \setminus P^I)$.

Given first-order interpretations I, J, J', 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 $J' \not\leq_I J$.

We now give a definition of the minimal change update semantics but in difference to (Winslett 1990), we use a specific vocabulary which is closer to the setting of this paper. In particular, we define the semantics of updating an initial theory $\mathcal{S}$ by an ABox $\mathcal{A}$ in the context of the TBox $\mathcal{T}$. The TBox is treated as static integrity constraints for the whole update process. The minimal change update semantics chooses those models of $\mathcal{T} \cup \mathcal{A}$ that are the closest w.r.t. the relation $\leq_I$ to some model I of $\mathcal{T} \cup \mathcal{S}$. Formally:

Definition 4 (*Winslett's minimal change update semantics*)
Let $\mathcal{S}$ be a first-order theory, $\mathcal{T}$ a TBox, $\mathcal{A}$ an ABox, I a first-order interpretation and M a set of first-order interpretations. We define:

$$\mathsf{incorporate}^{\mathcal{T}}(\mathcal{A}, I) = \{\, J \in \mathsf{mod}(\mathcal{T} \cup \mathcal{A}) \mid (\nexists J' \in \mathsf{mod}(\mathcal{T} \cup \mathcal{A}))(J' <_I J) \,\} \ ,$$
$$\mathsf{incorporate}^{\mathcal{T}}(\mathcal{A}, M) = \bigcup_{I \in M} \mathsf{incorporate}^{\mathcal{T}}(\mathcal{A}, I) \ ,$$
$$\mathsf{mod}(\mathcal{S} \oplus^{\mathcal{T}} \mathcal{A}) = \mathsf{incorporate}^{\mathcal{T}}(\mathcal{A}, \mathsf{mod}(\mathcal{T} \cup \mathcal{S})) \ .$$

If $\mathsf{mod}(\mathcal{S} \oplus^{\mathcal{T}} \mathcal{A})$ is nonempty, we call it the *minimal change update model of $\mathcal{S} \oplus^{\mathcal{T}} \mathcal{A}$*.

The previous definition can be naturally generalised to allow for sequences of ABoxes. Starting from the models of the original theory, for each ABox in the sequence we transform the set of models according to the minimal change update semantics defined above. The resulting set of models then determines the updated theory. Formally:

Definition 5 (*Update by a sequence of ABoxes*)
Let $\mathcal{S}$ be a first-order theory, $\mathcal{T}$ a TBox, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$ a sequence of ABoxes and M a set of first-order interpretations. We inductively define:

$$\mathsf{incorporate}^{\mathcal{T}}(\mathcal{A}, M) = \mathsf{incorporate}^{\mathcal{T}}((\mathcal{A}_2, \ldots, \mathcal{A}_n), \mathsf{incorporate}^{\mathcal{T}}(\mathcal{A}_1, M)) \enspace ,$$
$$\mathsf{mod}(\mathcal{S} \oplus^{\mathcal{T}} \mathcal{A}) = \mathsf{incorporate}^{\mathcal{T}}(\mathcal{A}, \mathsf{mod}(\mathcal{T} \cup \mathcal{S})) \enspace .$$

If $\mathsf{mod}(\mathcal{S} \oplus^{\mathcal{T}} \mathcal{A})$ is nonempty, we call it the *minimal change update model of* $\mathcal{S} \oplus^{\mathcal{T}} \mathcal{A}$.

3 Hybrid Update Operator

Turning to the formal part of our proposal, our aim is to propose a semantics for a program $\mathcal{P}$ updated by a sequence of ABoxes $(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$ in the context of a TBox $\mathcal{T}$. We assume program $\mathcal{P}$ to be finite and ground, a common assumption when dealing with reasoning under the stable model semantics.

We follow a path similar to how the stable models of normal logic programs were originally defined (Gelfond and Lifschitz 1988), and start by defining how a definite program can be updated by a sequence of ABoxes, and only afterwards deal with programs containing default negation.

As with the least model of a definite logic program, our resulting model is the least fixed point of an immediate consequence operator. Our operator is in a way similar to the usual immediate consequence operator $T_{\mathcal{P}}$ commonly used to draw consequences from a logic program $\mathcal{P}$. The crucial difference between $T_{\mathcal{P}}$ and our operator is that in the latter, the consequences are subsequently updated by the sequence of ABoxes $\mathcal{A}$ using the classical update operator. Formally:

Definition 6 (*Updating immediate consequence operator* $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$)
Let $\mathcal{P}$ be a finite ground definite program, $\mathcal{T}$ a TBox and $\mathcal{A}$ a sequence of ABoxes. We define the operator $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$ for any $M \subseteq \mathcal{I}$ as follows[2]:

$$T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}(M) = \mathsf{mod}(\{\, H^*(r) \mid r \in \mathcal{P} \wedge M \models B(r) \,\} \oplus^{\mathcal{T}} \mathcal{A})$$

An important property of an immediate consequence operator is *continuity* because it guarantees the existence of a least fixed point and also provides a way of computing this least fixed point (using the Kleene Fixed Point Theorem). The $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$ operator satisfies the condition of continuity:

Proposition 7 (*Continuity of* $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$)
Let $\mathcal{P}$ be a finite ground definite program, $\mathcal{T}$ a TBox and $\mathcal{A}$ a sequence of ABoxes. Then $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$ is a continuous function on the complete partial order of all subsets of $\mathcal{I}$ with the least element $\mathcal{I}$.

Now we can define a *minimal change dynamic stable model* of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$, where $\mathcal{P}$ is a definite program, as the least fixed point of $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$:

[2] Recall that $M \models B(r)$ holds if and only if M is an S5 model of every modal atom in $B(r)$ (full definition is on page 551).

Definition 8 (*Minimal change dynamic stable model for definite programs*)
Let $\mathcal{P}$ be a finite ground definite program, $\mathcal{T}$ a TBox and $\mathcal{A}$ a sequence of ABoxes. We say an MKNF interpretation M is a *minimal change dynamic stable model of* $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if it is the least fixed point of $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$.

Notice that for every definite program $\mathcal{P}$ and each sequence of ABoxes $\mathcal{A}$, $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ has either no minimal change dynamic stable model (when the least fixed point of $T_{\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}}$ is empty), or exactly one minimal change dynamic stable model.

In order to deal with default negation in the bodies of rules, we use the Gelfond-Lifschitz transformation which was used to define the stable models of a normal logic program (Gelfond and Lifschitz 1988). We do this by defining the definite program $\mathcal{P}^M$ which is the result of performing the Gelfond-Lifschitz transformation on $\mathcal{P}$ – rules from $\mathcal{P}$ with a negative body that is in conflict with M are discarded, while for all the other rules, their negative bodies are discarded. Then $\mathcal{P}^M$ is updated by $\mathcal{A}$ using the above definition for definite logic programs and if the result is identical to M, then M is given the status of a *minimal change dynamic stable model*. Hence, the resulting operator can be used to update an arbitrary normal logic program by a sequence of ABoxes.

Definition 9 (*Minimal change dynamic stable model*)
Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}$ a TBox, $\mathcal{A}$ a sequence of ABoxes and M an MKNF interpretation. We say M is a *minimal change dynamic stable model of* $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if M is a minimal change dynamic stable model of $\mathcal{P}^M \oplus^{\mathcal{T}} \mathcal{A}$ where

$$\mathcal{P}^M = \left\{ H(r) \subset B^+(r) \;\middle|\; r \in \mathcal{P} \land M \models B^-(r) \right\} \ .$$

The minimal change dynamic stable models can be used to define a consequence relation from $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ where $\mathcal{P}$ is a finite ground program, $\mathcal{T}$ is a TBox and $\mathcal{A}$ a sequence of ABoxes. We offer a definition which adopts a skeptical approach to inference, credulous and other definitions may be obtained similarly.

Definition 10 (*Consequence relation*)
Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}$ a TBox, $\mathcal{A}$ a sequence of ABoxes and ϕ an MKNF sentence. We say that $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ *entails* ϕ, written $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A} \models \phi$, if and only if $M \models \phi$ for all minimal change dynamic stable models M of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$.

We now demonstrate the defined update semantics on a simple example:

Example 11
Consider the following TBox $\mathcal{T}$ and program $\mathcal{P}$:

$$\mathcal{T}: \quad A \equiv B \sqcup C \tag{2}$$

$$NegA \equiv \neg A \tag{3}$$

$$D \equiv \neg A \sqcap \exists P^{-}.A \tag{4}$$

$$\mathcal{P}: \quad NegA(X) \leftarrow not\, A(X). \tag{5}$$

$$P(X,Y) \leftarrow A(X), E(Y), not\, E(X). \tag{6}$$

TBox assertions (2) and (3) together with rule (5) define the concept A as a union of concepts B and C and they make this concept interpreted under CWA instead of OWA, i.e. whenever for some constant c we cannot conclude that $A(c)$ is true, the rule (5) infers

$NegA(c)$ and by (3) we obtain $\neg A(c)$. Assertion (4) defines concept D as those members d of $\neg A$ for which there exists some c from A with $P(c,d)$. Rule (6) infers the relation $P(c,d)$ whenever c is in A but not in E and d is in E.

Given the initial definitions, an update by $\mathcal{A}_1 = \{ A(c) \}$ now yields[3]

$$\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}_1 \models \{ A(c), \neg A(d) \} \quad .$$

A further update by $\mathcal{A}_2 = \{ \neg B(c) \}$ introduces a possibility of $A(c)$ not being true in case $B(c)$ was true before and $C(c)$ was false. Since A is interpreted under the closed world assumption, we can now conclude that $A(c)$ is false:

$$\mathcal{P} \oplus^{\mathcal{T}} (\mathcal{A}_1, \mathcal{A}_2) \models \{ \neg A(c), \neg B(c), \neg A(d) \}$$

Consider now the update $\mathcal{A}_3 = \{ C(c) \wedge E(d) \}$. Given (2), this reinstates $A(c)$. Furthermore, rule (6) can now infer $P(c,d)$ and by (3) we obtain $D(d)$:

$$\mathcal{P} \oplus^{\mathcal{T}} (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3) \models \{ A(c), \neg B(c), C(c), \neg A(d), E(d), P(c,d), D(d) \}$$

In the next update $\mathcal{A}_4 = \{ E(c) \}$ we block the body of rule (6), which also prevents $D(d)$ from being inferred:

$$\mathcal{P} \oplus^{\mathcal{T}} (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4) \models \{ A(c), \neg B(c), C(c), \neg A(d), E(d), E(c) \}$$

The last update[4] $\mathcal{A}_5 = \{ \neg E(c) \wedge \neg P(c,d) \}$ illustrates how the conclusion of a rule may be overridden through the ABox updates – though the body of rule (6) is true, its head does not become true since it is in direct conflict with $\mathcal{A}_5$:

$$\mathcal{P} \oplus^{\mathcal{T}} (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_5) \models \{ A(c), \neg B(c), C(c), \neg A(d), E(d), \neg E(c), \neg P(c,d) \}$$

4 Properties and Relations

In this section we investigate a number of formal properties of the defined operator. The first property guarantees that every minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ is a model of $\mathcal{A}$. This is known as the *principle of primacy of new information* (Dalal 1988).

Proposition 12 (*Primacy of new information*)
Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}$ a TBox, $\mathcal{A}$ an ABox and M a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$. Then $M \models \mathcal{A}$.

The second property guarantees that our operator is syntax-independent w.r.t. the TBox and the updating ABox. This is a desirable property as it shows that providing equivalent TBoxes and updating by equivalent ABoxes always produces the same result. It is inherited from the classical minimal change update operator.

[3] In the example we assume that the rules are grounded using all constants explicitly mentioned in the knowledge base. In this case there are only two: c and d.

[4] Updating ABoxes could, of course, be more complex since arbitrary concept expressions may be used (e.g. $(\exists P.C)(c)$). Here, due to limited space, we keep the example very simple.

Proposition 13 (*Syntax independence*)
Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}, \mathcal{T}'$ be TBoxes such that $\mathsf{mod}(\mathcal{T}) = \mathsf{mod}(\mathcal{T}')$, $\mathcal{A}, \mathcal{A}'$ be ABoxes such that $\mathsf{mod}(\mathcal{A}) = \mathsf{mod}(\mathcal{A}')$ and M be an MKNF interpretation. Then M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if and only if M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}'} \mathcal{A}'$.

The following proposition relates the hybrid update operator to the static MKNF semantics of hybrid knowledge bases. It gives sufficient conditions for the static and dynamic semantics to coincide. In particular, the sufficient condition requires that for any set of consequences S of program $\mathcal{P}$ in the context of a model M, updating S by $\mathcal{A}$ in the context of $\mathcal{T}$ has the same effect as making an intersection of the models of S with the models of $\mathcal{A}$ and $\mathcal{T}$.

Proposition 14 (*Relation to Hybrid MKNF*)
Let $\mathcal{P}$ be a finite ground program, $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ an ontology with TBox $\mathcal{T}$ and ABox $\mathcal{A}$ and M an MKNF interpretation such that for every subset S of the set $\{H^*(r) | r \in \mathcal{P} \wedge M \models B(r)\}$ the following condition is satisfied:

$$\mathsf{mod}(S \oplus^{\mathcal{T}} \mathcal{A}) = \mathsf{mod}(S \cup \mathcal{O}) \enspace .$$

Then M is an MKNF model of $\langle \mathcal{O}, \mathcal{P} \rangle$ if and only if M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$.

The precondition of this proposition is satisfied, for example, when predicates appearing in heads of $\mathcal{P}$ do not appear in the ontology $\mathcal{O}$. An important subcase of this is when $\mathcal{O}$ is empty because then the proposition implies that the minimal change dynamic stable models of $\mathcal{P} \oplus^{\emptyset} \emptyset$ are exactly the MKNF models of $\mathcal{P}$. Since the MKNF semantics generalises the stable model semantics (Lifschitz 1991), the minimal change dynamic stable models of $\mathcal{P} \oplus^{\emptyset} \emptyset$ also coincide with the stable models of $\mathcal{P}$. In other words, our operator properly generalises stable models.

Corollary 15 (*Generalisation of stable models*)
Let $\mathcal{P}$ be a finite ground program. Then M is a stable model of $\mathcal{P}$ if and only if M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\emptyset} \emptyset$.

Turning to relations with the minimal change update operator, we show that updating any logic program that can be equivalently translated into first-order logic has the same effect as updating the translated first-order theory using the minimal change update operator. Hence, our update operator generalises the classical minimal change update operator.

Proposition 16 (*Generalisation of the minimal change update operator*)
Let $\mathcal{P}$ be a finite ground program containing only facts, $\mathcal{T}$ a TBox, $\mathcal{A}$ a sequence of ABoxes and M an MKNF interpretation. Then M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if and only if M is a minimal change update model of $\mathcal{S}_{\mathcal{P}} \oplus^{\mathcal{T}} \mathcal{A}$ where $\mathcal{S}_{\mathcal{P}} = \{ p \mid \mathbf{K}\, p \in \mathcal{P} \}$.

Another property that our operator inherits from the classical minimal change update operator is that empty ABoxes in the updating sequence do not influence the resulting models. Similarly, updating an empty program simply yields the set of all first-order models of $\mathcal{T} \cup \mathcal{A}$. These last two properties ensure that empty program and updates cannot influence the resulting models under our update operator[5].

Proposition 17 (*Indifference to empty updates*)
Let $\mathcal{P}$ be a finite ground program, $\mathcal{T}$ be a TBox and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$ a sequence of ABoxes (where $n \geq 1$). Let $\mathcal{A}' = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_i, \emptyset, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n)$ for some $i \in \{ 0, 1, 2, \ldots, n \}$. Then an MKNF interpretation M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}$ if and only if M is a minimal change dynamic stable model of $\mathcal{P} \oplus^{\mathcal{T}} \mathcal{A}'$.

Proposition 18 (*Updating an empty program*)
Let $\mathcal{T}$ be a TBox, $\mathcal{A}$ an ABox and M an MKNF interpretation. Then M is a minimal change dynamic stable model of $\emptyset \oplus^{\mathcal{T}} \mathcal{A}$ if and only if $M = \mathsf{mod}(\mathcal{T} \cup \mathcal{A})$.

Relation to Katsuno and Mendelzon's postulates

In the following we briefly discuss the relation of our operator to Katsuno and Mendelzon's postulates for updates of propositional knowledge bases formulated in (Katsuno and Mendelzon 1991). Each propositional knowledge base over a finite language can be represented by a single propositional formula and the result of the update can also be represented as a propositional formula. The eight desirable properties of an update operator $\diamond$ are as follows:

KM 1: $\phi \diamond \psi$ implies ψ.

KM 2: If ϕ implies ψ, then $\phi \diamond \psi$ is equivalent to ϕ.

KM 3: If both ϕ and ψ are satisfiable, then $\phi \diamond \psi$ is satisfiable.

KM 4: If ϕ_1 is equivalent to ϕ_2 and ψ_1 is equivalent to ψ_2, then $\phi_1 \diamond \psi_1$ is equivalent to $\phi_2 \diamond \psi_2$.

KM 5: $(\phi \diamond \psi) \wedge \chi$ implies $\phi \diamond (\psi \wedge \chi)$.

KM 6: If $\phi \diamond \psi_1$ implies ψ_2 and $\phi \diamond \psi_2$ implies ψ_1, then $\phi \diamond \psi_1$ is equivalent to $\phi \diamond \psi_2$.

KM 7: If for each atom p either ϕ implies p or ϕ implies $\neg p$, then $(\phi \diamond \psi_1) \wedge (\phi \diamond \psi_2)$ implies $\phi \diamond (\psi_1 \vee \psi_2)$.

KM 8: $(\phi_1 \vee \phi_2) \diamond \psi$ is equivalent to $(\phi_1 \diamond \psi) \vee (\phi_2 \diamond \psi)$.

In order to examine these postulates in our setting, we restrict our attention to a finite propositional language. In order to interpret the postulates in our setting, we need to define the semantics of a number of notions used in them. Let $\mathcal{P}, \mathcal{P}_1, \mathcal{P}_2$ be programs, $\mathcal{T}$ a TBox and $\alpha, \alpha_1, \alpha_2$ be propositional formulae representing ABox updates. We need to discuss and define, at least:

1. When does $\mathcal{P} \oplus^{\mathcal{T}} \alpha_1$ imply α_2? (used in KM 1 and KM 6)
2. When does $\mathcal{P}$ imply α? (used in KM 2 and KM 7)
3. When is $\mathcal{P}_1 \oplus^{\mathcal{T}} \alpha$ equivalent to $\mathcal{P}_2$? (used in KM 2)
4. When is $\mathcal{P}$ satisfiable? (used in KM 3)
5. When is $\mathcal{P} \oplus^{\mathcal{T}} \alpha$ satisfiable? (used in KM 3)

[5] Perhaps surprisingly, as shown in (Leite 2003), these two properties are violated by many update operators in the context of Logic Programming.

6. When is $\mathcal{P}_1$ equivalent to $\mathcal{P}_2$? (used in KM 4)
7. When is $\mathcal{P}_1 \oplus^{\mathcal{T}} \alpha_1$ equivalent to $\mathcal{P}_2 \oplus^{\mathcal{T}} \alpha_2$? (used in KM 4 and KM 6)
8. What is the semantics of $(\mathcal{P} \oplus^{\mathcal{T}} \alpha_1) \wedge \alpha_2$? (used in KM 5)
9. What is the semantics of $(\mathcal{P} \oplus^{\mathcal{T}} \alpha_1) \wedge (\mathcal{P} \oplus^{\mathcal{T}} \alpha_2)$? (used in KM 7)
10. What is the semantics of $\mathcal{P}_1 \vee \mathcal{P}_2$? (used in KM 8)

Most of these questions can be answered in multiple different ways while some of them are hard to provide answers to at all. In the following, we suggest ways of answering most of these questions and then analyse whether our operator satisfies the corresponding postulates.

Question 1. can be answered using the consequence relation from Def. 10. A similar consequence relation can be defined using stable models to answer question 2. A simple answer to question 3. is to say that $\mathcal{P}_1 \oplus^{\mathcal{T}} \alpha$ is equivalent to $\mathcal{P}_2$ if the set of minimal change dynamic stable models of $\mathcal{P}_1 \oplus^{\mathcal{T}} \alpha$ is equal to the set of stable models of $\mathcal{P}_2$. Regarding questions 4. and 5., we can say that $\mathcal{P}$ is satisfiable if it has at least one stable model and $\mathcal{P} \oplus^{\mathcal{T}} \alpha$ is satisfiable if it has at least one minimal change dynamic stable model. Question 6. can be answered similarly as question 3. by comparing the sets of minimal change dynamic stable models of $\mathcal{P} \oplus^{\mathcal{T}} \alpha_1$ and $\mathcal{P} \oplus^{\mathcal{T}} \alpha_2$. Finally, question 7. can be answered by comparing the sets of stable models of $\mathcal{P}_1$ and $\mathcal{P}_2$ or by using strong equivalence (Lifschitz et al. 2001). Providing reasonable answers to the remaining questions requires more investigation, so, for now, we do not further examine postulates KM 5, KM 7 and KM 8.

Turning to the rest of the postulates, we note that our operator adheres to KM 1, which was proved in Proposition 12. The same is not the case with postulate KM 2, as shown by the following counterexample. Consider the program

$$\begin{aligned} \mathcal{P}: \quad & p \leftarrow \mathit{not}\, q. \qquad & r \leftarrow q, \mathit{not}\, r. \\ & q \leftarrow \mathit{not}\, p. \qquad & r \leftarrow p. \end{aligned} \tag{7}$$

and an update $\alpha = r$. The only stable model of $\mathcal{P}$ is the maximal S5 model M of $\{ p, r \}$. Clearly, $M \models \alpha$. But $\mathcal{P} \oplus^{\mathcal{T}} \alpha$ has another minimal change dynamic stable M', which is the maximal S5 model of $\{ q, r \}$ and so is not equivalent to $\mathcal{P}$.

In fact, this behaviour is inherited from the stable semantics for logic programs which does not satisfy the very similar property of *cumulativity* (Makinson 1988; Dix 1995). Hence, it is expectable that KM 2 is never satisfied by any update semantics that properly generalises the stable model semantics.

A similar situation arises with postulate KM 3 because the stable model semantics allows to express integrity constraints, and these may easily be broken by an update. For example, the program $\mathcal{P} = \{ p \leftarrow q, \mathit{not}\, p. \}$, updated by $\alpha = q$, of which both are satisfiable, does not allow for any minimal change dynamic stable model. It is not clear how an integrity constraint should be updated because, once it is a part of the knowledge base, which is assumed to be a correct representation of the world, it should not be violated, and no new information should have the power to override it. Or should it? That is another open research question worth investigating.

Postulate KM 4 is partially formulated in Proposition 13, which shows that updating by equivalent ABoxes produces the same result. The other half amounts to proving that updating equivalent logic programs by the same ABox also produces equivalent results.

For the two notions of program equivalence that we proposed above, this property does not hold. As a counterexample take $\mathcal{P}_1 = \{ p., q. \}$ and $\mathcal{P}_2 = \{ p., q \leftarrow p. \}$ which have the same answer sets and are also strongly equivalent. An update by $\alpha = \neg p$, produces different results for $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, which we believe is in accord with intuitions regarding these two programs. It may be the case that for different notions of program equivalence that better suit our scenario, such as the update equivalence of logic programs proposed in (Leite 2003), this property holds. Further investigation is needed to answer this question.

Finally, postulate KM 6 is also not satisfied by the operator. As a counterexample we can take the program $\mathcal{P}$ defined in (7), $\alpha_1 = r$ and $\alpha_2 = p \vee q$. Then $\mathcal{P} \oplus^{\mathcal{T}} \alpha_1$ has two minimal change dynamic stable models: $M_1 = \mathsf{mod}(\{ p, r \})$ and $M_2 = \mathsf{mod}(\{ q, r \})$. Hence, $P \oplus^{\mathcal{T}} \alpha_1 \models \alpha_2$. Furthermore, $\mathcal{P} \oplus^{\mathcal{T}} \alpha_2$ has only one minimal change dynamic stable model which is M_1 and consequently $\mathcal{P} \oplus^{\mathcal{T}} \alpha_2 \models \alpha_1$. However, $\mathcal{P} \oplus^{\mathcal{T}} \alpha_1$ is not equivalent to $\mathcal{P} \oplus^{\mathcal{T}} \alpha_2$.

5 Conclusion and Future Work

As seen, our operator properly generalises the two main ingredients that it is motivated by – the stable model semantics of normal logic programs (Corollary 15) and the minimal change update operator (Proposition 16). The failure of our operator to satisfy many of Katsuno and Mendelzon's postulates is not surprising. A wide range of classical update and revision postulates was already studied in the context of rule updates, only to find that many of them were inappropriate for characterising plausible rule update operators (Eiter et al. 2002). Furthermore, in (Slota and Leite 2010) we show that even under the SE model semantics, which is strictly more expressive than stable models semantics, update operators satisfying only some of the basic Katsuno and Mendelzon's postulates necessarily violate the property of support which is at the core of most logic programming semantics. The search for desirable properties of hybrid update operators is an interesting future research area.

There are also many more properties still to be examined, among them decidability as well as complexity of reasoning. Since we cannot expect the operator to perform any better than the stable model semantics and the classical update operator it is based on, its tractable approximations need to be defined and examined. The well-founded semantics for logic programs (Gelder et al. 1991) and its version for hybrid MKNF knowledge bases (Alferes et al. 2009) constitute crucial starting points. The recent research on ontology evolution (see (Flouris et al. 2008) for a survey) can help design tractable update operators which, at the same time, offer the necessary functionality to be interesting for use in practice.

In this paper, the TBox was considered static and was treated in the same way as integrity constraints in (Winslett 1990). This approach to handling integrity constraints in the context of updates has been criticized in the literature (Herzig and Rifi 1999; Herzig 2005), as in certain cases it does not provide the expected results. However, the proposed solutions are defined only for the propositional case and a preliminary examination showed that their treatment of equivalences, such as the TBox definitions used in Example 11, is not always the expected one. Further investigation is needed to find suitable solutions to these problems in the context of ontology updates. Furthermore, in truly dynamic environments, the TBox should also be allowed to be updated. We believe that finding appropriate update operators for ontologies is still a largely open research question.

The large body of work on rule updates (Leite 2003; Alferes et al. 2005), and more recently (Delgrande et al. 2008), also needs to be exploited in the attempts to define an update operator that can deal with the evolution of both rules and ontologies.

Finally, while incorporating new knowledge in a knowledge base is important, the complementary task of removing a certain piece of information is also important. Hence, hybrid erasure operators should be studied and related to hybrid update operators. The work on erasure (Giacomo et al. 2007) in description logics as well as forgetting in both description logics (Wang et al. 2009) and logic programs (Eiter and Wang 2008) should be the starting points of this research.

To conclude, in this paper, to the best of our knowledge, we proposed the first update operator for hybrid knowledge bases. We deal with a constrained but interesting scenario in which a TBox and nonmonotonic rules represent static knowledge, policies, norms and default preferences, and the evolving ABox represents the open and dynamic environment. We illustrated the behaviour of our operator on a simple example. The operator can be used in realistic scenarios where the general notions and rules are relatively fixed, and individuals tend to change their state frequently. This is the case of many real life institutions where stakeholders change their state on a regular basis while the general rules and structures change only occasionally.

We proved a number of properties of our operator, among which its relations with the theories it was based on, such as the stable model semantics for logic programs (Gelfond and Lifschitz 1988), the MKNF semantics for hybrid knowledge bases (Motik and Rosati 2007) and Winslett's minimal change update operator (Winslett 1990).

We believe that this new area of research brings exciting new problems to solve and bridges a number of existing research areas. It will certainly provide useful results for many applications and perhaps even contribute to finding further philosophical insights into how human knowledge evolves.

References

ALCHOURRÓN, C. E., GÄRDENFORS, P., AND MAKINSON, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic 50,* 2, 510–530.

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., KNORR, M., AND SWIFT, T. 2009. Queries to hybrid MKNF knowledge bases through oracular tabling. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*. 1–16.

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.

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.

BERNERS-LEE, T., HENDLER, J., AND LASSILA, O. 2001. The semantic web. *Scientific American 284,* 5, 28–37.

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., 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: I. strong properties. *Fundamenta Informaticae 22,* 3, 227–255.

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.

EITER, T. AND WANG, K. 2008. Semantic forgetting in answer set programming. *Artificial Intelligence 172,* 14, 1644–1672.

FLOURIS, G., MAKANATAS, D., KONDYLAKIS, H., PLEXOUSAKIS, D., AND ANTONIOU, G. 2008. Ontology change: classification and survey. *The Knowledge Engineering Review 23,* 2, 117–152.

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, Seattle, Washington, 1070–1080.

GIACOMO, G. D., LENZERINI, M., POGGI, A., AND ROSATI, R. 2007. On the approximation of instance level update and erasure in description logics. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*. AAAI Press, Vancouver, British Columbia, Canada, 403–408.

HERZIG, A. 2005. On updates with integrity constraints. In *Belief Change in Rational Agents: Perspectives from Artificial Intelligence, Philosophy, and Economics*, J. P. Delgrande, J. Lang, H. Rott, and J.-M. Tallon, Eds. Dagstuhl Seminar Proceedings, vol. 05321. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

HERZIG, A. AND RIFI, O. 1999. Propositional belief base update and minimal change. *Artificial Intelligence 115,* 1, 107–138.

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, Berlin, 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.

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, 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.

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.

MAKINSON, D. 1988. General theory of cumulative inference. In *Proceedings of the 2nd International Workshop on Non-Monotonic Reasoning (NMR'88)*, R. Michael, J. De Kleer, M. L. Ginsberg, and E. Sandewall, Eds. Lecture Notes in Computer Science, vol. 346. Springer, Grassau, FRG, 1–18.

MAREK, V. W. AND TRUSZCZYNSKI, M. 1998. Revision programming. *Theoretical Computer Science 190,* 2, 241–277.

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.

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. 2010. On semantic update operators for answer-set programs. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010).* To appear.

WANG, K., WANG, Z., TOPOR, R. W., PAN, J. Z., AND ANTONIOU, G. 2009. Concept and role forgetting in $\mathcal{ALC}$ ontologies. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, Eds. Lecture Notes in Computer Science, vol. 5823. Springer, Chantilly, VA, USA, 666–681.

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, NY, USA.

ZHANG, Y. AND FOO, N. Y. 2005. A unified framework for representing logic program updates. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, M. M. Veloso and S. Kambhampati, Eds. AAAI Press / The MIT Press, Pittsburgh, Pennsylvania, USA, 707–713.