A Faithful Integration of Description Logics with Logic Programming*

Boris Motik
Department of Computer Science
University of Manchester, UK

Riccardo Rosati
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza", Italy

Abstract

Integrating description logics (DL) and logic programming (LP) would produce a very powerful and useful formalism. However, DLs and LP are based on quite different principles, so achieving a seamless integration is not trivial. In this paper, we introduce *hybrid MKNF knowledge bases* that faithfully integrate DLs with LP using the logic of Minimal Knowledge and Negation as Failure (MKNF) [Lifschitz, 1991]. We also give reasoning algorithms and tight data complexity bounds for several interesting fragments of our logic.

1 Introduction

Integrating description logics (DLs) and logic programming (LP) in a common framework would produce a very powerful formalism; for example, LP rules could be used to model constraints and exceptions over DL knowledge bases. Adding a rule layer on top of the DL-based Web Ontology Language (OWL) is currently the central task in the development of the Semantic Web language stack, and the Rule Interchange Format (RIF) working group of the World Wide Web Consortium (W3C) is currently working on standardizing such a language. However, DLs and LP are based on fundamentally different assumptions: DLs are fragments of first-order logic and employ open-world semantics, whereas LP provides negation-as-failure under closed-world semantics.

Several approaches to integrating DLs and LP were proposed recently. [Rosati, 2006] proposes an approach that interprets the predicates from the DL knowledge base under open-world and the predicates occurring only in the LP rules under closed-world semantics. [Eiter *et al.*, 2004] propose a loose coupling of DLs and LP by extending the LP rules with special atoms interpreted as queries to a DL knowledge base.

In this paper, we introduce *hybrid MKNF knowledge bases*, which integrate an arbitrary description logic $\mathcal{DL}$[1] with disjunctive logic programs and negation-as-failure. This integration is *faithful* in the sense that it provides exactly the same consequences as $\mathcal{DL}$ and LP, respectively, if the other component is empty. Unlike the existing approaches, our approach allows the user to freely switch between open- and closed-world views on arbitrary predicates from $\mathcal{DL}$ and LP. By means of an example, we argue that our logic produces intuitive and practically useful consequences.

We base our approach on the logic of Minimal Knowledge and Negation as Failure (MKNF), developed by [Lifschitz, 1991] to unify several major approaches to nonmonotonic reasoning. To obtain intuitive consequences for our hybrid logic, we modify certain technical aspects of MKNF regarding the universe of discourse and equality.

Function-free logic programs are usually decidable if the rules are safe, and many decidable DLs exist. However, we show that these restrictions on $\mathcal{DL}$ and LP are not sufficient to obtain a decidable hybrid formalism, due to an interaction between $\mathcal{DL}$ and negation-as-failure. To obtain decidability, we apply the well-known notion of *DL-safety* [Motik *et al.*, 2005], which makes the rules applicable only to individuals known by name. We present several reasoning algorithms for the cases of general, positive (i.e., without negation-as-failure), nondisjunctive and positive, nondisjunctive and stratified, and nondisjunctive (but not necessarily stratified) DL-safe rules. Finally, we present tight data complexity bounds for each case and show that combining DLs with LP often does not increase the data complexity of LP.

Our approach is related to several nonmonotonic extensions of DLs. [Donini *et al.*, 2002] propose an MKNF-based extension of the DL $\mathcal{ALC}$, but without LP rules and with unique name assumption (UNA). [Baader and Hollunder, 1995] propose an extension of DLs with DL-safe default rules, which are subsumed by our approach due to our treatment of equality and the well-known embedding of default logic into MKNF [Lifschitz, 1991]. Finally, [Bonatti *et al.*, 2006] present an extension of DLs with circumscription that allows for nonmonotonic reasoning on unnamed individuals, but only for unary predicates if decidability is desired.

We leave the detailed proofs to [Motik and Rosati, 2006] and present here only proof sketches.

2 Preliminaries

Description Logics. Our approach is applicable to any first-order fragment $\mathcal{DL}$ satisfying these conditions: (*i*) each knowledge base $\mathcal{O} \in \mathcal{DL}$ can be translated into a formula

*This work was partially funded by the EPSRC project REOL (EP/C537211/1) and the EU project TONES (FET FP6-7603).

[1] Our approach is actually applicable to any first-order fragment; however, our work is motivated primarily by knowledge representation, so we call this fragment a description logic.

$\pi(\mathcal{O})$ of function-free first-order logic with equality, (*ii*) it supports *ABoxes*—assertions of the form $P(a_1, \ldots, a_n)$ for P a predicate and a_i constants of $\mathcal{DL}$, and (*iii*) satisfiability checking and instance checking (i.e., checking entailments of the form $\mathcal{O} \models P(a_1, \ldots, a_n)$) are decidable. We assume familiarity with the standard DL notation [Baader *et al.*, 2003].

MKNF. The first-order version of MKNF [Lifschitz, 1991] can be understood as a variant of the first-order modal logic S5 with a preference relation on models that implements the nonmonotonic semantics. The syntax of MKNF is obtained by extending first-order logic with modal operators $\mathbf{K}$ and $\mathbf{not}$. A formula $P(t_1, \ldots, t_n)$, where t_i are terms, is a *first-order atom*. For φ an MKNF formula, $\mathbf{K}\,\varphi$ and $\mathbf{not}\,\varphi$ are *modal* $\mathbf{K}$- and $\mathbf{not}$-*atoms*, respectively; φ is *ground* if it does not contain variables; φ is *positive* if it does not contain $\mathbf{not}$; and $\varphi[t/x]$ is the formula obtained from φ by replacing all free occurrences of the variable x with the term t.

We assume that, apart from the constants used in MKNF formulae, we have a countably infinite supply of constants not used in the formulae; with $\triangle$ we denote the Herbrand universe of such a signature. Also, we assume that the signature contains a special equality predicate $\approx$ that is interpreted as a congruence relation on $\triangle$. An *MKNF structure* is a triple (I, M, N), where I is a Herbrand first-order interpretation over $\triangle$, and M and N are nonempty sets of Herbrand first-order interpretations over $\triangle$. Satisfiability of closed MKNF formulae in an MKNF structure (I, M, N) is defined as follows, for A a first-order atom:

$(I, M, N) \models A$	iff A is true in I
$(I, M, N) \models \neg\varphi$	iff $(I, M, N) \not\models \varphi$
$(I, M, N) \models \varphi_1 \wedge \varphi_2$	iff $(I, M, N) \models \varphi_1$ and $(I, M, N) \models \varphi_2$
$(I, M, N) \models \exists x : \varphi$	iff $(I, M, N) \models \varphi[\alpha/x]$ for some $\alpha \in \triangle$
$(I, M, N) \models \mathbf{K}\,\varphi$	iff $(J, M, N) \models \varphi$ for all $J \in M$
$(I, M, N) \models \mathbf{not}\,\varphi$	iff $(J, M, N) \not\models \varphi$ for some $J \in N$

The symbols true, false, $\vee$, $\forall$, and $\subset$ (material implication) are interpreted as usual. An *MKNF interpretation* M is a nonempty set of first-order interpretations over $\triangle$. Let φ and ψ be closed MKNF formulae. An MKNF interpretation M is an S5 *model* of φ, written $M \models \varphi$, if $(I, M, M) \models \varphi$ for each $I \in M$; M is an *MKNF model* of φ if (*i*) M is an S5-model of φ and (*ii*) $(I', M', M) \not\models \varphi$ for each $M' \supset M$ and some $I' \in M'$ (so-called *preference semantics* of MKNF); finally, φ entails ψ, written $\varphi \models_{\mathsf{MKNF}} \psi$, if $(I, M, M) \models \psi$ for each MKNF model M of φ and $I \in M$.

3 Hybrid MKNF Knowledge Bases

Definition 3.1. *Let $\mathcal{O}$ be a DL knowledge base. A first-order function-free atom $P(t_1, \ldots, t_n)$ such that P is $\approx$ or it occurs in $\mathcal{O}$ is called a* DL-atom*; all other atoms are called* non-DL-atoms*. An* MKNF rule *r has the following form, where H_i, B_i^+, and B_i^- are first-order function-free atoms:*

$$\mathbf{K}\,H_1 \vee \ldots \vee \mathbf{K}\,H_n \leftarrow \mathbf{K}\,B_1^+, \ldots, \mathbf{K}\,B_m^+, \mathbf{not}\,B_1^-, \ldots, \mathbf{not}\,B_k^- \quad (1)$$

The sets $\{\mathbf{K}\,H_i\}$, $\{\mathbf{K}\,B_i^+\}$, and $\{\mathbf{not}\,B_i^-\}$ are called the rule head*, the* positive body*, and the* negative body*, respectively. A rule r is* nondisjunctive *if $n = 1$; r is* positive *if $k = 0$; r is a* fact *if $m = k = 0$; r is* safe *if all variables in r occur in a positive body atom. A* program *$\mathcal{P}$ is a finite set of MKNF rules. A* hybrid MKNF knowledge base *$\mathcal{K}$ is a pair $(\mathcal{O}, \mathcal{P})$.*

We define the semantics of $\mathcal{K}$ by translating it into a first-order MKNF formula as follows:

Definition 3.2. *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a hybrid MKNF knowledge base. We extend π to r, $\mathcal{P}$, and $\mathcal{K}$ as follows, where $\mathbf{x}$ is the vector of the free variables of r:*

$$\pi(r) = \forall \mathbf{x} : (\mathbf{K}\,H_1 \vee \ldots \vee \mathbf{K}\,H_n \subset \mathbf{K}\,B_1^+ \wedge \ldots \wedge \mathbf{K}\,B_m^+ \wedge \mathbf{not}\,B_1^- \wedge \ldots \wedge \mathbf{not}\,B_k^-)$$
$$\pi(\mathcal{P}) = \textstyle\bigwedge_{r \in \mathcal{P}} \pi(r) \qquad \pi(\mathcal{K}) = \mathbf{K}\,\pi(\mathcal{O}) \wedge \pi(\mathcal{P})$$

$\mathcal{K}$ is satisfiable *if and only if an MKNF model of $\pi(\mathcal{K})$ exists, and $\mathcal{K}$* entails *a closed MKNF formula ψ, written $\mathcal{K} \models \psi$, if and only if $\pi(\mathcal{K}) \models_{\mathsf{MKNF}} \psi$.*

MKNF as defined by [Lifschitz, 1991] considers arbitrary models, whereas we consider only Herbrand models (see Section 2). We introduce this restriction to obtain an intuitive logic that is compatible with both DLs and LP. Namely, under the semantics by Lifschitz, $\mathbf{K}\,A(a) \not\models_{\mathsf{MKNF}} \exists x : \mathbf{K}\,A(x)$, since $\mathbf{K}\,A(a)$ does not fix the interpretation of a in $\triangle$. To remedy that, we interpret all constants by themselves, which is standard in data management applications [Donini *et al.*, 2002; Reiter, 1992]. In order to ensure that our model is large enough, we assume a countably infinite supply of constants. Hence, we consider only infinite models of $\mathcal{O}$ in which different constants are interpreted as different objects.

An equality-free first-order formula is satisfiable in an arbitrary model iff it is satisfiable in a Herbrand model with an infinite supply of constants not occurring in the formula [Fitting, 1996, Theorem 5.9.4]. Hence, without equality we cannot distinguish arbitrary from Herbrand models. For compatibility with DLs, we need $\approx$, which we clearly cannot interpret as identity, so we interpret it as a congruence. Now a first-order formula with equality is satisfiable in a model with "true equality" iff it is satisfiable in a model where $\approx$ is an ordinary predicate interpreted as a congruence [Fitting, 1996, Theorem 9.3.9]. Hence, our approach is fully compatible with DLs: if $\mathcal{P} = \emptyset$, then $\mathcal{K} \models \varphi$ iff $\mathcal{O} \models \varphi$ for any first-order formula φ. Similarly, our logic is fully compatible with LP: for a ground atom α, if $\mathcal{O} = \emptyset$, then $\mathcal{K} \models \mathbf{K}\,\alpha$ iff $\mathcal{P}$ entails α under stable model semantics [Lifschitz, 1991].

We believe that our treatment of equality matches the common intuition behind negation-as-failure. In the version of MKNF by [Lifschitz, 1991], $\varphi = \mathbf{K}\,A(a) \wedge \neg\,\mathbf{not}\,A(b)$ has a model with a singleton universe in which a and b are the same, so φ entails $a \approx b$. We consider this quite unintuitive: we did not say that a and b are the same, so we expect them to be different. Because of these problems, the semantics of LP is commonly based on Herbrand models. We follow this practice, so in our approach φ is MKNF unsatisfiable, and it can be made satisfiable by explicitly stating $a \approx b$.

In [Motik and Rosati, 2006] we allow the rules to contain both modal and nonmodal atoms, in order to generalize known extensions of DLs with rules such as SWRL or $\mathcal{DL}$+log [Rosati, 2006]. However, we also show that general rules can be converted into rules with only modal atoms.

Furthermore, mixing modal and nonmodal atoms produces an unintuitive semantics, so it is good practice to use either fully nonmodal or fully modal rules.

DL-Safety. Making the rules safe usually suffices to make function-free LP decidable; however, we show next that this does not hold for MKNF rules. Note that MKNF rules contain modal atoms, so existing undecidability results for first-order combinations of DLs with rules are not directly applicable.

Theorem 3.3. *For $\mathcal{K}$ a safe hybrid MKNF knowledge base and A a ground atom, checking whether $\mathcal{K} \models A$ is undecidable if $\mathcal{DL}$ allows us to express an axiom $\top \sqsubseteq C$.*

Namely, $\top \sqsubseteq C$ makes C equivalent to the infinite set $\triangle$, so all elements from $\triangle$ can be accessed in the rules even if the rules are safe. In other words, safety is now not a sufficient condition for domain independence. To ensure decidability, we apply the well-known DL-safety restriction, which makes the rules applicable only to individuals known by name in the ABox; for an in-depth discussion, see [Motik *et al.*, 2005].

Definition 3.4. *An MKNF rule r is* DL-safe *if every variable in r occurs in at least one non-DL-atom $\mathbf{K}\,B$ in the body of r. A hybrid MKNF knowledge base $\mathcal{K}$ is* DL-safe *if all its rules are DL-safe.*

Note that DL-atoms are interpreted in the same way as non-DL-atoms; DL-safety merely provides a syntactic restriction ensuring decidability. For a hybrid MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, let $\mathcal{P}_G$ be obtained from $\mathcal{P}$ by replacing in each rule all variables with all constants from $\mathcal{K}$ in all possible ways; the knowledge base $\mathcal{K}_G = (\mathcal{O}, \mathcal{P}_G)$ is called a *ground instantiation* of $\mathcal{K}$.

Lemma 3.5. *For $\mathcal{K}$ a DL-safe hybrid MKNF knowledge base, $\mathcal{K}_G$ the ground instantiation of $\mathcal{K}$, and ψ a ground MKNF formula. Then, $\mathcal{K} \models \psi$ iff $\mathcal{K}_G \models \psi$.*

4 Example

Consider determining the car insurance premium based on various information about the driver. By convention, DL-predicates start with an uppercase and non-DL-predicates with a lowercase letter. Let $\mathcal{K}$ be the following hybrid MKNF knowledge base (the predicate p means "person"):

(2) $$\mathit{NotMarried} \equiv \neg \mathit{Married}$$

(3) $$\mathit{NotMarried} \sqsubseteq \mathit{HighRisk}$$

(4) $$\exists \mathit{Spouse}.\top \sqsubseteq \mathit{Married}$$

(5) $$\mathbf{K}\,\mathit{NotMarried}(x) \leftarrow \mathbf{K}\,p(x), \mathbf{not}\,\mathit{Married}(x)$$

(6) $$\mathbf{K}\,\mathit{Discount}(x) \leftarrow \mathbf{K}\,\mathit{Spouse}(x, y), \mathbf{K}\,p(x), \mathbf{K}\,p(y)$$

Let us now assert $p(\mathit{John})$. Under first-order semantics, some models contain $\mathit{Married}(\mathit{John})$ and others contain $\mathit{NotMarried}(\mathit{John})$. However, (5) applies the closed-world assumption to $\mathit{Married}$: by default, people are assumed not to be married. Since $\mathit{Married}(\mathit{John})$ does not hold in all models of $\mathcal{O}$, the rule (5) "fires" and it derives $\mathit{NotMarried}(\mathit{John})$; with (3), this implies $\mathit{HighRisk}(\mathit{John})$.

The preference semantics of MKNF is strongly related to the notion of the Gelfond-Lifschitz reduct used to define stable models. Let M_1 and M_2 be MKNF interpretations consisting of first-order interpretations I such that $I \models \mathit{Married}(\mathit{John})$ and $I \models \mathit{NotMarried}(\mathit{John})$, respectively, and let $M_3 = M_1 \cup M_2$. Clearly, we have $M_1 \subset M_3$, $M_1 \models \mathbf{K}\,\mathit{Married}(\mathit{John})$ and $M_3 \not\models \mathbf{K}\,\mathit{Married}(\mathit{John})$; intuitively, M_1 contains "more knowledge" than M_3. The preference semantics of MKNF gives the following semantics to **not**: an MKNF model M of an MKNF formula φ is the minimal knowledge justified by the values of **not**-atoms. In our example, assuming that $\mathbf{not}\,\mathit{Married}(\mathit{John})$ is true implies other consequences, such as $\mathbf{K}\,\mathit{NotMarried}(\mathit{John})$; now each such consequence is justified in the sense that it belongs to the minimal knowledge entailed by the knowledge base $\mathcal{K}$ in which each **not**-atom is replaced with its value.

To understand how open- and closed-world reasoning interact in our formalism, let us assert $\exists \mathit{Spouse}.\top(\mathit{Bill})$ and $p(\mathit{Bill})$. Now Bill can be married to different people in different models; however, (4) ensures that $\mathit{Married}(\mathit{Bill})$ holds in all models. The precondition of (5) is thus not satisfied, so we derive neither $\mathit{NotMarried}(\mathit{Bill})$ nor $\mathit{HighRisk}(\mathit{Bill})$. In a way, reasoning in $\mathcal{O}$ is performed under open-world semantics, but the modal operators allow us to put on "closed-world glasses" and consider the consequences in all models.

Finally, let us assert $\mathit{Spouse}(\mathit{Bob}, \mathit{Ann})$, $p(\mathit{Bob})$, and $p(\mathit{Ann})$. By (6) we can now derive $\mathit{Discount}(\mathit{Bob})$; in contrast, we cannot derive $\mathit{Discount}(\mathit{Bill})$. Namely, the rule (6) differs from the first-order implication (4) in that it requires $\mathit{Spouse}(x, y)$ to be known—that is, it must hold in all first-order models. The spouse of Bill is different in different models, so $\mathcal{O} \not\models \mathit{Spouse}(\mathit{Bill}, y)$ for any value of y, and the rule (6) would not "fire" even if it were not DL-safe.

This example cannot be expressed in existing combinations of DLs and LP: the rules of [Eiter *et al.*, 2004] cannot derive new DL facts, and, in the approach by [Rosati, 2006], the DL predicates are interpreted under open-world semantics and thus cannot occur under **not**.

5 Reasoning Algorithms

Given a nonground hybrid MKNF knowledge base $\mathcal{K}$, all our algorithms first compute the ground knowledge base $\mathcal{K}_G$ to obtain an MKNF theory without modal operators under quantifiers. For a program with variables, this step is exponential. However, even for nondisjunctive datalog the combined complexity is higher than data complexity by an exponential factor [Dantsin *et al.*, 2001], so this is to be expected of our logic as well. Therefore, for an MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, we just consider the *data complexity*, which is measured in the size of the facts in $\mathcal{P}$ and the size of the ABox of $\mathcal{O}$. We denote the data complexity of reasoning in $\mathcal{DL}$ with $\mathcal{C}$, and set $\mathcal{E} = \mathrm{NP}$ if $\mathcal{E} \subseteq \mathrm{NP}$ and $\mathcal{E} = \mathcal{C}$ otherwise.

For simplicity, we write $\mathcal{K}$, $\mathcal{O}$, and $\mathcal{P}$ instead of $\pi(\mathcal{K})$, $\pi(\mathcal{O})$, and $\pi(\mathcal{P})$. The MKNF models of $\mathcal{K}_G$ are clearly infinite, so we must devise a convenient finite representation for them. We adopt an approach already used by [Rosati, 1999]: we represent an MKNF model M using a first-order formula φ such that $M = \{I \mid I \models \varphi\}$. We formalize this as follows:

Definition 5.1. *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a hybrid MKNF knowledge base. The set of $\mathbf{K}$-atoms of $\mathcal{K}$, written $\mathsf{KA}(\mathcal{K})$, is the smallest set that contains (*i*) all $\mathbf{K}$-atoms of $\mathcal{P}_G$, and (*ii*) a modal atom $\mathbf{K}\,\xi$ for each modal atom $\mathbf{not}\,\xi$ occurring in $\mathcal{P}_G$.*

For a subset P of $\mathsf{KA}(\mathcal{K})$, the objective knowledge of P *is the formula* $\mathsf{ob}_{\mathcal{K},P} = \mathcal{O} \cup \bigcup_{\mathbf{K}\,\xi \in P} \xi$. *A partition (P, N) of $\mathsf{KA}(\mathcal{K})$ is* consistent *if $\mathsf{ob}_{\mathcal{K},P} \not\models \xi$ for each $\mathbf{K}\,\xi \in N$.*

Let φ be an MKNF formula and (P, N) a partition of $\mathsf{KA}(\mathcal{K})$. The formula $\varphi[\mathbf{K}, P, N]$ is obtained from φ by replacing each $\mathbf{K}\,\xi$ with true *if $\mathbf{K}\,\xi \in P$ and with* false *otherwise; $\varphi[\mathbf{not}, P, N]$ is obtained from φ by replacing each $\mathbf{not}\,\xi$ with* true *if $\mathbf{K}\,\xi \in N$ and with* false *otherwise; finally, $\varphi[P, N] = \varphi[\mathbf{K}, P, N][\mathbf{not}, P, N]$.*

For a set of modal atoms S, S_{DL} is the subset of DL-atoms of S, $\widehat{S} = \{\xi \mid \mathbf{K}\,\xi \in S\}$, and $\widehat{S}_{DL} = \widehat{S'}$ for $S' = S_{DL}$.

An MKNF model is strongly related to a particular partition of $\mathsf{KA}(\mathcal{K})$:

Definition 5.2. *An MKNF interpretation M* induces *the partition (P, N) of $\mathsf{KA}(\mathcal{K})$ if $\mathbf{K}\,\xi \in P$ implies $M \models \mathbf{K}\,\xi$ and $\mathbf{K}\,\xi \in N$ implies $M \not\models \mathbf{K}\,\xi$.*

The following key lemma shows that each MKNF model M of $\mathcal{K}_G$ can be represented as a partition of $\mathsf{KA}(\mathcal{K})$:

Lemma 5.3. *Let M be an MKNF model of $\mathcal{K}_G$ and (P, N) a partition of $\mathsf{KA}(\mathcal{K})$ induced by M. Then, M is equal to the set of interpretations $M' = \{I \mid I \models \mathsf{ob}_{\mathcal{K},P}\}$.*

Hence, to find an MKNF model of $\mathcal{K}_G$, we need to find a partition (P, N) of $\mathsf{KA}(\mathcal{K})$. We can do it in different ways for different types of rules.

The General Case. As for disjunctive datalog, in the general case we must guess a partition (P, N) of $\mathsf{KA}(\mathcal{K})$. This is captured by Algorithm 1.

Theorem 5.4. *Let $\mathcal{K}$ be a DL-safe hybrid MKNF knowledge base and $\psi = (\neg)\,\mathbf{K}\,A$ for A a ground atom. Then,* not-entails-DL$(\mathcal{K}, \psi)$ *returns* true *iff $\mathcal{K} \not\models \psi$, and it runs with data complexity $\mathcal{E}^{\mathcal{E}}$.*

Proof sketch. ($\Rightarrow$) If not-entails-DL$(\mathcal{K}, \psi)$ returns true, a partition (P, N) of $\mathsf{KA}(\mathcal{K})$ satisfying all conditions of Algorithm 1 exists. We show that $M = \{I \mid I \models \mathsf{ob}_{\mathcal{K},P}\}$ is an MKNF model of $\mathcal{K}_G$. By Condition (3), (P, N) is consistent on all DL-atoms and, by Condition (4), it is consistent on all non-DL-atoms (for a non-DL-predicate Q, $\mathsf{ob}_{\mathcal{K},P} \models Q(a_1, \ldots, a_n)$ can hold only if $Q(b_1, \ldots, b_n) \in \widehat{P}$ and $\mathsf{ob}_{\mathcal{K},P} \models a_i \approx b_i$ for all i). By Condition (2), $M \neq \emptyset$. By Condition (1), $(I, M, M) \models \mathcal{P}_G[P, N]$ for each $I \in M$; furthermore, the values of all $\mathbf{K}$-atoms in M are determined by (P, N), so $(I, M, M) \models \mathcal{P}_G$. Clearly, $(I, M, M) \models \mathcal{O}$, so $(I, M, M) \models \mathcal{K}_G$. Assume now that M is not an MKNF model—that is, an MKNF interpretation M' exists such that $M' \supset M$ and $(I', M', M) \models \mathcal{K}_G$ for each $I' \in M'$. Then, M' induces the partition (P'', N'') of $\mathsf{KA}(\mathcal{K})$. Because $M' \models \mathbf{K}\,\xi$ implies $M \models \mathbf{K}\,\xi$, we have $P'' \subset P$, so we can represent (P'', N'') as $(P', N \cup N')$ for (P', N') a partition of P with $N' \neq \emptyset$. By Lemma 5.3, $M' = \{I \mid I \models \mathsf{ob}_{\mathcal{K},P'}\}$. Because $M' \neq \emptyset$, $\mathsf{ob}_{\mathcal{K},P'}$ is satisfiable, so Condition (b) is invalidated; $(I', M', M) \models \gamma[P', N \cup N']$ for each $I' \in M'$, which invalidates Condition (a); finally, $(P', N \cup N')$ is clearly consistent, which invalidates Conditions (c) and (d). Hence, Condition (5) could not hold for $(P', N \cup N')$, which is a contradiction; hence, M is an MKNF model of $\mathcal{K}_G$. Finally, Condition (6) ensures $M \not\models \psi$, so $\mathcal{K}_G \not\models \psi$. The ($\Leftarrow$) direction is analogous.

Algorithm 1 Entailment in General hybrid MKNF KBs

Algorithm: not-entails-DL$(\mathcal{K}, \psi)$
Input:
$\mathcal{K} = (\mathcal{O}, \mathcal{P})$: a DL-safe hybrid MKNF knowledge base
ψ : a ground formula $(\neg)\,\mathbf{K}\,A$
Output:
true if $\mathcal{K} \not\models \psi$; false otherwise

let $\mathcal{K}_G$ be the ground instantiation of $\mathcal{K}$
if a partition (P, N) of $\mathsf{KA}(\mathcal{K}_G) \cup \{\mathbf{K}\,A\}$ exists such that
1. $\mathcal{P}_G[P, N]$ evaluates to true, and
2. $\mathcal{O} \cup \widehat{P}_{DL}$ is satisfiable, and
3. $\mathcal{O} \cup \widehat{P}_{DL} \not\models \xi$ for each $\mathbf{K}\,\xi \in N_{DL}$, and
4. **for each** $Q(a_1, \ldots, a_n) \in \widehat{N}$ and $Q(b_1, \ldots, b_n) \in \widehat{P}$, we have $\mathcal{O} \cup \widehat{P}_{DL} \not\models a_i \approx b_i$ for some $1 \leq i \leq n$
5. **for** $\gamma = \mathcal{P}_G[\mathbf{not}, P, N]$ and each partition (P', N') of P such that $N' \neq \emptyset$
   (a) $\gamma[P', N \cup N']$ evaluates to false, or
   (b) $\mathcal{O} \cup \widehat{P}'_{DL}$ is unsatisfiable, or
   (c) $\mathcal{O} \cup \widehat{P}'_{DL} \models \xi$ for some $\mathbf{K}\,\xi \in N'_{DL}$, or
   (d) **for some** $Q(a_1, \ldots, a_n) \in \widehat{N}'$ and $Q(b_1, \ldots, b_n) \in \widehat{P}'$, we have $\mathcal{O} \cup \widehat{P}'_{DL} \models a_i \approx b_i$ for all $1 \leq i \leq n$
6. one of the following conditions holds:
   (i) $\psi = \mathbf{K}\,A$ and $\mathbf{K}\,A \notin P$, or
   (ii) $\psi = \neg\,\mathbf{K}\,A$ and $\mathbf{K}\,A \in P$

then return true; **otherwise return** false

$\mathcal{K}_G$ can be computed in polynomial time if the size of the nonground rules in $\mathcal{P}$ is bounded. A partition (P, N) can be guessed and Condition (1) can be checked in time polynomial in the size of $\mathcal{K}_G$. Checking Conditions (2)–(4) and (6) requires a polynomial number of calls to an oracle running in $\mathcal{C}$, so all these steps can be performed in $\mathcal{E}$. Disproving Condition (5) requires guessing a partition (P', N') and a polynomial number of calls to an oracle running in $\mathcal{C}$, so it can be performed in $\mathcal{E}$. Hence, validating Condition (5) can be performed in co$\mathcal{E}$, and the algorithm runs in $\mathcal{E}^{\mathcal{E}}$. □

Positive Programs. For $\mathcal{K}$ a positive hybrid MKNF knowledge base, $\mathcal{K} \models \mathbf{K}\,A$ iff $M \models \mathbf{K}\,A$ for each S5-model M of $\mathcal{K}$: for each such M, since $\mathcal{K}$ is positive, an MKNF model M' of $\mathcal{K}$ exists such that $M' \subseteq M$. (This is analogous to the case of positive disjunctive datalog.) In other words, for positive queries, we do not need to ensure the preference semantics of MKNF. Let not-entails-DL$^+(\mathcal{K}, \mathbf{K}\,A)$ be an algorithm that is the same as Algorithm 1, only without Condition (5). By adapting the proof of Theorem 5.4, we get the following:

Theorem 5.5. *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a positive DL-safe hybrid MKNF knowledge base and A a ground atom. Then,* not-entails-DL$^+(\mathcal{K}, \mathbf{K}\,A)$ *returns* true *iff $\mathcal{K} \not\models \mathbf{K}\,A$, and it runs with data complexity $\mathcal{E}$.*

Positive Nondisjunctive Programs. If $\mathcal{K}$ is nondisjunctive and positive, we can construct (P, N) deterministically in a bottom-up fashion. We can easily show that $M_1 \models \mathcal{K}$ and $M_2 \models \mathcal{K}$ implies $M_1 \cup M_2 \models \mathcal{K}$ for all MKNF interpretations M_1 and M_2, which immediately implies the following:

Theorem 5.6. *Each positive DL-safe nondisjunctive hybrid MKNF knowledge base $\mathcal{K}$ has at most one MKNF model.*

Definition 5.7. *For $\mathcal{K}$ a positive nondisjunctive DL-safe hybrid MKNF knowledge base, $R_\mathcal{K}$, $D_\mathcal{K}$, and $T_\mathcal{K}$ are the operators defined on the subsets of $\mathsf{KA}(\mathcal{K})$ as follows:*

$$R_\mathcal{K}(S) = S \cup \{\mathbf{K}\, H \mid \mathcal{K}_G \text{ contains a rule of the form (1) such that } \mathbf{K}\, B_i \in S \text{ for each } 1 \leq i \leq n\}$$

$$D_\mathcal{K}(S) = \{\mathbf{K}\, \xi \mid \mathbf{K}\, \xi \in \mathsf{KA}(\mathcal{K}) \text{ and } \mathcal{O} \cup \widehat{S}_{DL} \models \xi\} \cup \{\mathbf{K}\, Q(b_1, \ldots, b_n) \mid \mathbf{K}\, Q(a_1, \ldots, a_n) \in S \setminus S_{DL} \text{ and } \mathcal{O} \cup \widehat{S}_{DL} \models s_i \approx b_i \text{ for } 1 \leq i \leq n\}$$

$$T_\mathcal{K}(S) = R_\mathcal{K}(S) \cup D_\mathcal{K}(S)$$

The operator $T_\mathcal{K}$ is monotonic on the lattice of subsets of $\mathsf{KA}(\mathcal{K})$ (i.e., $S \subseteq S'$ implies $T_\mathcal{K}(S) \subseteq T_\mathcal{K}(S')$). Namely, $R_\mathcal{K}$ is monotonic similarly as this is the case for datalog, and $D_\mathcal{K}$ is monotonic because first-order logic is monotonic. Hence, by Knaster-Tarski's theorem, $T_\mathcal{K}$ has the least fixpoint, which we denote with $T^\omega_\mathcal{K}$. It is now easy to show the following:

Theorem 5.8. *Let $\mathcal{K}$ be a positive nondisjunctive DL-safe hybrid MKNF knowledge base and $M = \{I \mid I \models \mathsf{ob}_{\mathcal{K}, T^\omega_\mathcal{K}}\}$. Then, (*i*) if $M \neq \emptyset$, then M is the single MKNF model of $\mathcal{K}$; (*ii*) if $\mathcal{K}$ has an MKNF model, this model is equal to M; and (*iii*) the data complexity of computing $T^\omega_\mathcal{K}$ is in $\mathrm{P}^\mathcal{C}$.*

Proof sketch. (*i*) Clearly, $M \models \mathbf{K}\, \mathcal{O}$ and, since $T^\omega_\mathcal{K}$ is a fixpoint of $T_\mathcal{K}$, $M \models \mathcal{P}_G$. If M were not an MKNF model, an MKNF interpretation $M' \supset M$ such that $(I', M', M) \models \mathcal{K}_G$ for each $I' \in M'$ would exist and it would induce a partition (P', N') of $\mathsf{KA}(\mathcal{K})$ such that $P' \subset P$. Clearly, $T_\mathcal{K}(P') = P'$ (otherwise, either $M' \not\models r$ for some $r \in \mathcal{P}_G$, or (P', N') is not consistent), but then $T^\omega_\mathcal{K}$ is not the minimal fixpoint of $T_\mathcal{K}$. The proof for (*ii*) is analogous. (*iii*) As for ordinary datalog, $R_\mathcal{K}(S)$ can be computed in polynomial time. The number of atoms in $\mathsf{KA}(\mathcal{K})$ is linear in the size of $\mathcal{P}_G$, so computing $D_\mathcal{K}(S)$ requires a polynomial number of calls to an oracle running in $\mathcal{C}$. Finally, the number of iterations is bounded by the size of $\mathsf{KA}(\mathcal{K})$, which implies the claim. □

For entailment checking, $\mathcal{K} \models \mathbf{K}\, A$ iff $\mathsf{ob}_{\mathcal{K}, T^\omega_\mathcal{K}} \models A$, and $\mathcal{K} \models \neg\, \mathbf{K}\, A$ iff $\mathsf{ob}_{\mathcal{K}, T^\omega_\mathcal{K}} \not\models A$.

Stratified Programs. We first define a notion of stratification appropriate to MKNF programs.

Definition 5.9. *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a nondisjunctive hybrid MKNF knowledge base and $\lambda : \mathcal{P}_G \to \mathbb{N}^+$ a function assigning to each $r \in \mathcal{P}_G$ a positive integer $\lambda(r)$. For an integer k and $\bowtie \in \{<, \leq, >, \geq\}$, let $\mathsf{head}(\mathcal{K})^{\bowtie k}$ be the set of head atoms of those rules $r \in \mathcal{P}_G$ for which $\lambda(r) \bowtie k$. Then, λ is a* stratification *of $\mathcal{K}$ if these conditions hold for each $r \in \mathcal{P}_G$:*

- *For each body atom $\mathbf{K}\, \xi$ of r, each $P \subseteq \mathsf{head}(\mathcal{K})^{\leq \lambda(r)}$ such that $\mathsf{ob}_{\mathcal{K}, P} \not\models \xi$, and each $P' \subseteq \mathsf{head}(\mathcal{K})^{> \lambda(r)}$, either $\mathsf{ob}_{\mathcal{K}, P \cup P'} \not\models \xi$ or $\mathsf{ob}_{\mathcal{K}, P \cup P'}$ is unsatisfiable;*
- *For each body atom $\mathbf{not}\, \xi$ of r, each $P \subseteq \mathsf{head}(\mathcal{K})^{< \lambda(r)}$ such that $\mathsf{ob}_{\mathcal{K}, P} \not\models \xi$, and each $P' \subseteq \mathsf{head}(\mathcal{K})^{\geq \lambda(r)}$, either $\mathsf{ob}_{\mathcal{K}, P \cup P'} \not\models \xi$ or $\mathsf{ob}_{\mathcal{K}, P \cup P'}$ is unsatisfiable.*

$\mathcal{K}$ is stratified *if a stratification λ of $\mathcal{K}$ exists. A stratification λ partitions $\mathcal{P}$ into* strata *$\sigma_i = \{r \mid \lambda(r) = i\}$; the sequence of strata $\sigma_1, \ldots, \sigma_n$ is often identified with λ and is also called a stratification.*

Stratification ensures that deriving an atom $\mathbf{K}\, \xi$ in a stratum σ_i does not change the values of the $\mathbf{K}$- and $\mathbf{not}$-atoms from strata $< i$ and $\leq i$, respectively. For example, consider a knowledge base $\mathcal{K}$ where $\mathcal{DL}$ is propositional logic, $\mathcal{O} = (r \equiv p \vee q) \wedge (s \equiv \neg q)$, and $\mathcal{P}$ contains the rules $\mathbf{K}\, r \leftarrow \mathbf{not}\, p$ and $\mathbf{K}\, s \leftarrow \mathbf{K}\, r$. In ordinary datalog, $\mathcal{P}$ would be stratified: by evaluating the first rule we derive $\mathbf{K}\, r$, after which we evaluate the second rule and derive $\mathbf{K}\, s$. But now the objective knowledge is $r \wedge s \wedge \mathcal{O}$ and it implies $\neg q$, which invalidates the atom $\mathbf{K}\, p$ in the body of the first rule. Intuitively, Definition 5.9 ensures that such an interaction between $\mathcal{O}$ and the derived facts does not happen. Clearly, checking stratification is difficult in general. However, if $\mathcal{O}$ employs UNA and no rule in $\mathcal{P}$ contains a DL-atom in the head, then stratification of $\mathcal{P}$ can be checked as usual. This case is interesting because it allows to define complex constraints over DL knowledge bases. We now show how to compute models of a stratified program:

Definition 5.10. *Let $\sigma_1, \ldots, \sigma_k$ be a stratification of a DL-safe MKNF knowledge base $\mathcal{K}$. The sequence of subsets $U_0, \ldots, U_k$ of $\mathsf{KA}(\sigma)$ is inductively defined as $U_0 = \emptyset$ and, for $0 < i \leq k$, $U_i = T^\omega_{\chi_i}$ where $\chi_i = U_{i-1} \cup \sigma'_i$ and σ'_i is obtained from σ_i by replacing each $\mathbf{not}\, \xi$ with true if $U_{i-1} \not\models \xi$ and with false otherwise. Finally, let $U^\omega_\mathcal{K} = U_k$.*

The following theorem is analogous to the case of ordinary stratified datalog:

Theorem 5.11. *Let $\mathcal{K}$ be a stratified DL-safe MKNF knowledge base $\mathcal{K}$, and $M = \{I \mid I \models \mathsf{ob}_{\mathcal{K}, U^\omega_\mathcal{K}}\}$. Regardless of the stratification used to compute $U^\omega_\mathcal{K}$, the following claims hold: (*i*) if $M \neq \emptyset$, then M is an MKNF model of $\mathcal{K}$; (*ii*) if $\mathcal{K}$ has an MKNF model, then this model is equal to M; and (*iii*) the data complexity of computing $U^\omega_\mathcal{K}$ is in $\mathrm{P}^\mathcal{C}$.*

Nondisjunctive Nonstratified Programs. We now take $\mathcal{K}$ to be a nondisjunctive and nonstratified knowledge base. Then, the knowledge base $\gamma = \mathcal{P}_G[\mathbf{not}, P, N]$ from Algorithm 1 is nondisjunctive and positive, so we can use Theorem 5.8 to ensure the preference semantics of MKNF. We define $\mathsf{nondisjunctive\text{-}not\text{-}entails\text{-}DL}(\sigma, \psi)$ to be the same as Algorithm 1, but replace Condition (5) with $T^\omega_\gamma = P$.

Theorem 5.12. *For a nondisjunctive nonstratified hybrid MKNF knowledge base $\mathcal{K}$ and $\psi = (\neg)\, \mathbf{K}\, A$ with A a ground atom, the algorithm $\mathsf{nondisjunctive\text{-}not\text{-}entails\text{-}DL}(\mathcal{K}, \psi)$ returns true iff $\mathcal{K} \not\models \psi$, and it runs with data complexity $\mathcal{E}^{\mathrm{P}^\mathcal{C}}$.*

6 Data Complexity

We now determine the data complexity—the complexity measured in the sizes of the ABox of $\mathcal{O}$ and the facts of $\mathcal{P}$—of checking entailment for a hybrid MKNF knowledge base

Table 1: Data Complexity of Entailment Checking

	$\vee$	**not**	$\mathcal{DL} = \emptyset$	$\mathcal{DL} \in \mathrm{P}$	$\mathcal{DL} \in \mathrm{coNP}$
1	no	no	P	P	coNP
2	no	strat.	P	P	Δ_2^p
3	no	yes	coNP	coNP	Π_2^p
4	yes	no	coNP/Π_2^p	coNP/Π_2^p	coNP/Π_2^p
5	yes	yes	Π_2^p	Π_2^p	Π_2^p

$\mathcal{K}$. To obtain a precise characterization, we must make assumptions about the data complexity of checking entailment of ground literals in $\mathcal{DL}$. In [Hustadt *et al.*, 2005], it was shown that checking entailment of ground atoms in many very expressive DLs, such as $\mathcal{SHIQ}$, is data complete for coNP. Furthermore, there are expressive fragments, such as Horn-$\mathcal{SHIQ}$ [Hustadt *et al.*, 2005] or DL-lite [Calvanese *et al.*, 2006], for which data complexity of entailment is in P. Therefore, we analyze the complexity of MKNF knowledge bases for these two cases, and contrast them with the well-known results for logic programs without a DL knowledge base. Table 1 summarizes the results for complexity of checking $\mathcal{K} \models \psi$, for $\psi = (\neg)\,\mathbf{K}\,A$ with A a ground atom. All results are completeness results.

Existing results for the corresponding variants of datalog [Dantsin *et al.*, 2001] provide hardness results for all cases but for rows 2 and 3 with $\mathcal{DL} \in \mathrm{coNP}$. For these cases, we give hardness proofs in [Motik and Rosati, 2006]. In the first case, we present a reduction from DAGS(SAT) [Gottlob, 1995], and in the second one from 2-QBF.

The membership results are given by the algorithms from Section 5. Note that, for rows 1 and 2 and $\psi = \neg\,\mathbf{K}\,A$, since $\mathcal{K}$ has at most one model, $\mathcal{K} \models \neg\,\mathbf{K}\,A$ iff either $\mathcal{K}$ is unsatisfiable of $\mathcal{K} \not\models \mathbf{K}\,A$, so answering negative queries can be reduced to answering positive ones. Furthermore, in row 4 the complexity differs depending on whether we ask a positive or a negative query: in the first case, we can use Theorem 5.5, whereas in the second case we must use Theorem 5.4.

7 Conclusion

Based on the logic MKNF by [Lifschitz, 1991], we have developed the formalism of hybrid MKNF knowledge bases that provides for a faithful integration of an arbitrary description logic with logic programming. Our approach seamlessly integrates open- and closed-world reasoning, without requiring an a priori commitment to either paradigm. It is fully compatible with the first-order semantics of DLs and logic programming under stable model semantics. We have developed reasoning algorithms and identified tight complexity bounds for interesting fragments of our formalism. In future, we shall try to extend our approach to well-founded semantics, as this might provide better complexity in some practical cases. Furthermore, we shall implement our approach in the ontology management system KAON2.[2]

[2] http://kaon2.semanticweb.org/

References

[Baader and Hollunder, 1995] F. Baader and B. Hollunder. Embedding Defaults into Terminological Knowledge Representation Formalisms. *Journal of Automated Reasoning*, 14(1):149–180, 1995.

[Baader *et al.*, 2003] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, January 2003.

[Bonatti *et al.*, 2006] P. Bonatti, C. Lutz, and F. Wolter. Description Logics with Circumscription. In *Proc. KR 2006*, pages 400–410, Lake District, UK, 2006.

[Calvanese *et al.*, 2006] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data Complexity of Query Answering in Description Logics. In *Proc. KR 2006*, pages 260–270, Lake District, UK, 2006.

[Dantsin *et al.*, 2001] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[Donini *et al.*, 2002] F. M. Donini, D. Nardi, and R. Rosati. Description Logics of Minimal Knowledge and Negation as Failure. *ACM Transactions on Computational Logic*, 3(2):177–225, 2002.

[Eiter *et al.*, 2004] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining Answer Set Programming with Description Logics for the Semantic Web. In *Proc. KR 2004*, pages 141–151, Whistler, Canada, 2004.

[Fitting, 1996] M. Fitting. *First-Order Logic and Automated Theorem Proving, 2nd Edition*. Texts in Computer Science. Springer, 1996.

[Gottlob, 1995] G. Gottlob. NP Trees and Carnap's Modal Logic. *Journal of the ACM*, 42(2):421–457, 1995.

[Hustadt *et al.*, 2005] U. Hustadt, B. Motik, and U. Sattler. Data Complexity of Reasoning in Very Expressive Description Logics. In *Proc. IJCAI 2005*, pages 466–471, Edinburgh, UK, 2005.

[Lifschitz, 1991] V. Lifschitz. Nonmonotonic Databases and Epistemic Queries. In *Proc. IJCAI '91*, pages 381–386, Sydney, Australia, 1991.

[Motik and Rosati, 2006] B. Motik and R. Rosati. Closing Semantic Web Ontologies. Technical report, University of Manchester, UK, 2006.

[Motik *et al.*, 2005] B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with rules. *Journal of Web Semantics*, 3(1):41–60, 2005.

[Reiter, 1992] R. Reiter. What Should a Database Know? *Journal of Logic Programming*, 14(1–2):127–153, 1992.

[Rosati, 1999] R. Rosati. Reasoning about Minimal Belief and Negation as Failure. *Journal of Artificial Intelligence Research*, 11:277–300, 1999.

[Rosati, 2006] R. Rosati. $\mathcal{DL} + log$: A Tight Integration of Description Logics and Disjunctive Datalog. In *Proc. KR 2006*, pages 68–78, Lake District, UK, 2006.