Semantic and computational advantages of the safe integration of ontologies and rules

Riccardo Rosati

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
rosati@dis.uniroma1.it

Abstract. Description Logics (DLs) are playing a central role in ontologies and in the Semantic Web, since they are currently the most used formalisms for building ontologies. Both semantic and computational issues arise when extending DLs with rule-based components. In particular, integrating DLs with nonmonotonic rules requires to properly deal with two semantic discrepancies: (a) DLs are based on the Open World Assumption, while rules are based on (various forms of) Closed World Assumption; (b) The DLs specifically designed for the Semantic Web, i.e., OWL and OWL-DL, are not based on the Unique Name Assumption, while rule-based systems typically adopt the Unique Name Assumption. In this paper we present the following contributions: (1) We define *safe hybrid knowledge bases*, a general formal framework for integrating ontologies and rules, which provides for a clear treatment of the above semantic issues; (2) We present a reasoning algorithm and establish general decidability and complexity results for reasoning in safe hybrid KBs; (3) As a consequence of these general results, we close a problem left open in [18], i.e., decidability of OWL-DL with DL-safe rules.

1 Introduction

The integration of structured knowledge bases (KBs) and rules has recently received considerable attention in the research on ontologies and the Semantic Web (see e.g.,[15, 1]). Description Logics (DLs) [2] are playing a central role in this field, since they are currently the most used formalisms for building ontologies, and have been proposed as standard languages for the specification of ontologies in the Semantic Web [19].

Practically all the approaches in this field concern the study of description logic knowledge bases augmented with rules expressed in Datalog (and its nonmonotonic extensions). Many semantic and computational problems have emerged in this research area. Among them, we concentrate on the following main issues/goals:

(1) *OWA vs. CWA:* DLs are fragments of first-order logic (FOL), hence their semantics is based on the *Open World Assumption* (OWA) of classical logic, while rules are based on a *Closed World Assumption* (CWA), imposed by the different semantics for logic programming and deductive databases (which formalize various notions of information closure). How to integrate the OWA of DLs and the CWA of rules

in a "proper" way? I.e., how to merge monotonic and nonmonotonic components from a semantic viewpoint?

(2) *UNA vs. non-UNA:* some DLs, in particular the ones specifically tailored for the Semantic Web, i.e., OWL and OWL-DL, are not based on the *Unique Name Assumption* (UNA) (we recall that the UNA imposes that different terms denote different objects). On the other hand, the standard semantics of Datalog rules is based on the UNA (see e.g. [4] for a discussion on this semantic discrepancy). How to define a non-UNA-based semantics for DLs and rules? and most importantly, is it possible to reason under the non-UNA-based semantics by exploiting standard (i.e., UNA-based) Datalog engines?

(3) *decidability preservation:* as shown by the first studies in this field [16], decidability (and complexity) of reasoning is a crucial issue in systems combining DL KBs and Datalog rules. In fact, in general this combination does not preserve decidability, i.e., starting from a DL KB in which reasoning is decidable and a rule KB in which reasoning is decidable, reasoning in the KB obtained by integrating the two components may not be a decidable problem.

(4) *modularity of reasoning:* can reasoning in DL KBs augmented with rules be performed in a modular way, strongly separating reasoning about the structural component and reasoning about the rule component? This is a very desirable property, since it allows for defining reasoning techniques (and engines) on top of deductive methods (and implemented systems) developed separately for DLs [2] and for Datalog and its nonmonotonic extensions [8].

In this paper, we present an approach which addresses all the above aspects. In particular, we present *safe hybrid KBs*, which extend the framework of r-hybrid KBs presented in [21] to the treatment of KBs interpreted without the UNA. Safe hybrid KBs are constituted of a structural component, which can be expressed in any fragment of FOL (e.g., in a DL), and a relational component, corresponding to a disjunctive Datalog (Datalog$^{\neg\vee}$) program [7]. The way in which the two components interact is restricted to be *safe*. This notion of safe interaction follows (and extends) the ideas proposed in [5, 16, 18].

We prove that all the above listed goals are reached by safe hybrid KBs. More specifically:

- (1),(2) We show that safe hybrid KBs provide a clear formal treatment of the above semantic issues, i.e., the semantics of safe hybrid KBs does not assume unique names, and accounts for OWA on the structural component, and CWA on the relational component.
- (3) We establish decidability and complexity results for reasoning in safe hybrid KBs, which prove that, under very general conditions, the safe integration of two decidable components preserves decidability of reasoning.
- (4),(2) Our algorithm implies that reasoning in safe hybrid KBs can be done by strongly separating reasoning about the structural component and reasoning about the rule component. Furthermore, our algorithm allows for reasoning under the non-UNA-based semantics by exploiting reasoning methods and systems for standard, UNA-based, disjunctive Datalog.

– Moreover, as a consequence of these general results, we close a problem left open in [18], i.e., decidability of OWL-DL with DL-safe rules.

The paper is structured as follows. In Section 2 we define syntax and semantics of safe hybrid KBs. In Section 3 we study reasoning in safe hybrid KBs: we first define an algorithm for satisfiability of safe hybrid KBs, then address decidability and complexity of reasoning with safe hybrid KBs. We discuss related work in Section 4. Finally, we draw some conclusions in Section 5. Due to space limits, proofs of theorems are omitted in the present version of the paper.

2 Safe hybrid KBs

In this section we define syntax and semantics of safe hybrid KBs. We introduce a monotonic, first-order semantics and a nonmonotonic semantics based on stable models.

2.1 Syntax

We denote by $\mathcal{L}$ any subset of the language of function-free first-order logic with equality (for example, a description logic language) over an alphabet of predicates $\mathcal{A} = \mathcal{A}_P \cup \mathcal{A}_R$, with $\mathcal{A}_P \cap \mathcal{A}_R = \emptyset$, and an alphabet of constants $\mathcal{C}$. Every $p \in \mathcal{A}_P$ is called a *structural predicate*. We represent the special equality predicate by the binary predicate symbol *equal* (for ease of notation, in the paper we write equality in prefixed notation), and assume that *equal* is a structural predicate, i.e., it belongs to $\mathcal{A}_P$. An *atom* is an expression of the form $r(X)$, where r is a predicate in $\mathcal{A}$ of arity n and X is a n-tuple of variables and constants. If no variable symbol occurs in X, then $r(X)$ is called a *ground atom*.

Definition 1. *A* safe hybrid KB $\mathcal{H}$ *is a pair* $(\mathcal{T}, \mathcal{P})$*, where:*

- $\mathcal{T} \subseteq \mathcal{L}$ *and no predicate in* $\mathcal{A}_R$ *occurs in* $\mathcal{T}$. $\mathcal{L}$ *is called the* structural language of $\mathcal{H}$*;*
- $\mathcal{P}$ *is a* Datalog$^{\neg\vee}$ program *over the predicate alphabet* $\mathcal{A}$ *and the alphabet of constants* $\mathcal{C}$*, i.e., a set of Datalog*$^{\neg\vee}$ *rules where each rule* R *has the form*

$$p_1(X_1) \vee \ldots \vee p_n(X_n) \leftarrow r_1(Y_1), \ldots, r_m(Y_m), s_1(Z_1), \ldots, s_k(Z_k), \mathit{not}\, u_1(W_1), \ldots, \mathit{not}\, u_h(W_h)$$

such that $n \geq 0$, $m \geq 0$, $k \geq 0$, $h \geq 0$, *each* $p_i(X_i)$, $r_i(Y_i)$, $s_i(Z_i)$, $u_i(W_i)$ *is an atom and:*
  - *each* p_i *is a predicate from* $\mathcal{A}$*;*
  - *each* r_i, u_i *is a predicate from* $\mathcal{A}_R$*;*
  - *each* s_i *is a predicate from* $\mathcal{A}_P$*;*
  - (safeness condition) *each variable occurring in* R *must occur in one of the* r_i*'s.*

  If $n = 0$*, we call* R *a* constraint. *If, for all* $R \in \mathcal{P}$, $n \leq 1$, $\mathcal{P}$ *is called a* Datalog$^{\neg}$ program. *If, for all* $R \in \mathcal{P}$, $n \leq 1$ *and* $h = 0$, $\mathcal{P}$ *is called a* positive Datalog program. *If there are no occurrences of variable symbols in* $\mathcal{P}$, $\mathcal{P}$ *is called a* ground program.

Informally, $\mathcal{P}$ is a Datalog$^{\neg\vee}$ program with a special safeness condition: in each rule R, each variable occurring in R must occur in a positive atom in the body of R whose predicate is from $\mathcal{A}_R$, i.e., does not occur in $\mathcal{T}$. Notice that such a condition strengthens the standard Datalog range restriction condition on the use of variables in rules.

Thus, the structural component and the rule component share the predicates in $\mathcal{A}_P$ and the constants in $\mathcal{C}$, while the alphabet of predicates $\mathcal{A}_R$ is only used by $\mathcal{P}$.

2.2 Semantics

We now define two semantics for safe hybrid KBs: the first one relies on a first-order logic interpretation of both the structural and the rule component of the safe hybrid KB, while the second semantics provides a nonmonotonic meaning to rules. From now on, unless specified otherwise, we call *interpretation* a first-order interpretation of the predicates in $\mathcal{A}$ and the constants in $\mathcal{C}$. The notion of satisfaction of a first-order sentence (or a first-order theory) in a first-order interpretation is the standard one in first-order logic.

First-order semantics The first-order semantics of a safe hybrid KB consists of a classical first-order interpretation not only of the structural component, but also of the rule component of the safe hybrid KB. Formally, let R be the following Datalog$^{\neg\vee}$ rule:

$$\begin{aligned} R = p_1(X_1, c_1) \vee \ldots \vee p_n(X_n, c_n) \leftarrow\ & r_1(Y_1, d_1), \ldots, r_m(Y_m, d_m), \\ & s_1(Z_1, e_1), \ldots, s_k(Z_k, e_k), \\ & \mathit{not}\ u_1(W_1, f_1), \ldots, \mathit{not}\ u_h(W_h, f_h) \end{aligned} \quad (1)$$

where each X_i, Y_i, Z_i, W_i is a set of variables and each c_i, d_i, e_i, f_i is a set of constants. Then, *FO*(R) is the first-order sentence

$$\begin{aligned} &\forall \overline{x}_1, \ldots, \overline{x}_n, \overline{y}_1, \ldots, \overline{y}_m, \overline{z}_1, \ldots, \overline{z}_k, \overline{w}_1, \ldots, \overline{w}_h. \\ &\quad r_1(\overline{y}_1, d_1) \wedge \ldots \wedge r_m(\overline{y}_m, d_m) \wedge s_1(\overline{z}_1, e_1) \wedge \ldots \wedge s_k(\overline{z}_k, e_k) \wedge \\ &\quad \neg u_1(\overline{w}_1, f_1) \wedge \ldots \wedge \neg u_h(\overline{w}_h, f_h) \rightarrow p_1(\overline{x}_1, c_1) \vee \ldots \vee p_n(\overline{x}_n, c_n) \end{aligned}$$

Given a Datalog$^{\neg\vee}$ program $\mathcal{P}$, *FO*$(\mathcal{P})$ is the set of first-order sentences $\{$*FO*$(R) \mid R \in \mathcal{P}\}$.

A *FOL-model* of a safe hybrid KB $\mathcal{H}$ is an interpretation $\mathcal{I}$ such that $\mathcal{I}$ satisfies $\mathcal{T} \cup$ *FO*$(\mathcal{P})$. $\mathcal{H}$ is called *FOL-satisfiable* if it has at least a FOL-model.

Finally, we define skeptical entailment under the FOL semantics. A sentence $\varphi \in \mathcal{L}$ is *FOL-entailed* by $\mathcal{H}$, denoted by $\mathcal{H} \models_{FOL} \varphi$ iff, for each FOL-model $\mathcal{I}$ of $\mathcal{H}$, $\mathcal{I}$ satisfies φ.

Notice that the above first-order semantics of rules does not distinguish between negated atoms in the body and disjunction in the head of rules: e.g., according to such semantics, the rules $A \leftarrow B, \mathit{not}\ C$ and $A \vee C \leftarrow B$ have the same meaning.

Nonmonotonic semantics An alternative semantics to safe hybrid KBs is based on a nonmonotonic interpretation of the rule component, according to the notion of *stable* *model* [10]. This is the semantics commonly adopted in Disjunctive Logic Programming (DLP) and in Disjunctive Datalog [7]. We now formalize such a semantics in the framework of safe hybrid KBs.

Given an interpretation $\mathcal{I}$, we denote by $\mathcal{I}_R$ the projection of $\mathcal{I}$ to $\mathcal{A}_R$ and $\mathcal{C}$, i.e., $\mathcal{I}_R$ is obtained from $\mathcal{I}$ by restricting it to the interpretation of the predicates in $\mathcal{A}_R$ and the constants in $\mathcal{C}$. Analogously, we denote by $\mathcal{I}_P$ the projection of $\mathcal{I}$ to $\mathcal{A}_P$ and $\mathcal{C}$, and denote $\mathcal{I}$ as $\mathcal{I}_P \cup \mathcal{I}_R$.

The *ground instantiation of* $\mathcal{P}$ *with respect to* $\mathcal{C}$, denoted by *gr*$(\mathcal{P}, \mathcal{C})$, is the program obtained from $\mathcal{P}$ by replacing every rule R in $\mathcal{P}$ with the set of rules obtained by applying all possible substitutions of variables in R with constants in $\mathcal{C}$.

Given an interpretation $\mathcal{I}$ of an alphabet of predicates $\mathcal{A}' \subset \mathcal{A}$ and the constants $\mathcal{C}$, and a ground program $\mathcal{P}_g$ over the predicates in $\mathcal{A}$, the *projection of* $\mathcal{P}_g$ *with respect to* $\mathcal{I}$, denoted by $\Pi(\mathcal{P}_g, \mathcal{I})$, is the ground program obtained from $\mathcal{P}_g$ as follows. For each rule $R \in \mathcal{P}_g$:

- delete R if there exists an atom $r(t)$ in the head of R such that $r \in \mathcal{A}'$ and $t^{\mathcal{I}} \in r^{\mathcal{I}}$;
- delete each atom $r(t)$ in the head of R such that $r \in \mathcal{A}'$ and $t^{\mathcal{I}} \notin r^{\mathcal{I}}$;
- delete R if there exists an atom $r(t)$ in the body of R such that $r \in \mathcal{A}'$ and $t^{\mathcal{I}} \notin r^{\mathcal{I}}$;
- delete each atom $r(t)$ in the body of R such that $r \in \mathcal{A}'$ and $t^{\mathcal{I}} \in r^{\mathcal{I}}$;

Informally, the projection of $\mathcal{P}_g$ with respect to $\mathcal{I}$ corresponds to evaluating $\mathcal{P}_g$ with respect to $\mathcal{I}$, thus eliminating from $\mathcal{P}_g$ every atom whose predicate is interpreted in $\mathcal{I}$. Thus, when $\mathcal{A}' = \mathcal{A}_P$, all occurrences of structural predicates are eliminated in the projection of $\mathcal{P}_g$ with respect to $\mathcal{I}$, according to the evaluation in $\mathcal{I}$ of the atoms with structural predicates occurring in $\mathcal{P}_g$.

Then, we introduce the notions of minimal model and stable model of a Datalog$^{\neg\vee}$ program where the UNA is not adopted.[1] Given two interpretations $\mathcal{I}_1, \mathcal{I}_2$ of the set of predicates $\mathcal{A}$ and the set of constants $\mathcal{C}$, we write $\mathcal{I}_1 \subset_{\mathcal{A},\mathcal{C}} \mathcal{I}_2$ if (i) for each $p \in \mathcal{A}$ and for each tuple t of constants from $\mathcal{C}$, if $t^{\mathcal{I}_1} \in p^{\mathcal{I}_1}$ then $t^{\mathcal{I}_2} \in p^{\mathcal{I}_2}$, and (ii) there exist $p \in \mathcal{A}$ and tuple t of constants from $\mathcal{C}$ such that $t^{\mathcal{I}_1} \notin p^{\mathcal{I}_1}$ and $t^{\mathcal{I}_2} \in p^{\mathcal{I}_2}$.

Given a positive ground Datalog$^{\neg\vee}$ program $\mathcal{P}$ over an alphabet of predicates $\mathcal{A}_R$ and an interpretation $\mathcal{I}$, we say that $\mathcal{I}$ is a *minimal model* of $\mathcal{P}$ if $\mathcal{I}$ satisfies *FO*$(\mathcal{P})$ and there is no interpretation $\mathcal{I}'$ such that $\mathcal{I}'$ satisfies *FO*$(\mathcal{P})$ and $\mathcal{I}' \subset_{\mathcal{A}_R,\mathcal{C}} \mathcal{I}$.

Given a ground Datalog$^{\neg\vee}$ program $\mathcal{P}$ and an interpretation $\mathcal{I}$ for $\mathcal{P}$, the *GL-reduct* [10] of $\mathcal{P}$ with respect to $\mathcal{I}$, denoted by *GL*$(\mathcal{P}, \mathcal{I})$, is the positive ground program obtained from $\mathcal{P}$ as follows. For each rule $R \in \mathcal{P}$: (i) delete R if there exists a negated atom $not\, r(t)$ in the body of R such that $t^{\mathcal{I}} \in r^{\mathcal{I}}$; (ii) delete each negated atom $not\, r(t)$ in the body of R such that $t^{\mathcal{I}} \notin r^{\mathcal{I}}$.

Given a ground Datalog$^{\neg\vee}$ program $\mathcal{P}$ and an interpretation $\mathcal{I}$, $\mathcal{I}$ is a *stable model* for $\mathcal{P}$ iff $\mathcal{I}$ is a minimal model of *GL*$(\mathcal{P}, \mathcal{I})$.

Given a safe hybrid KB $\mathcal{H} = (\mathcal{T}, \mathcal{P})$, we say that an interpretation $\mathcal{I}$ is a *NM-model* for $\mathcal{H}$ if the following conditions hold: (i) $\mathcal{I}_P$ satisfies $\mathcal{T}$; (ii) $\mathcal{I}_R$ is a stable model for

[1] Observe that the notions of minimal model and stable model presented here slightly differs from the standard ones for Datalog$^{\neg\vee}$, since they are expressed in a more general framework in which unique names are not assumed. Consequently, the interpretation of constants must be considered in the definition of minimal and stable model.

$\Pi(gr(\mathcal{P},\mathcal{C}),\mathcal{I}_P)$. $\mathcal{H}$ is called *NM-satisfiable* (or simply *satisfiable*) if $\mathcal{H}$ has at least a *NM*-model.

Finally, we define skeptical entailment in safe hybrid KBs under the nonmonotonic semantics, which is analogous to the previous notion of entailment under the first-order semantics. We say that a sentence $\varphi \in \mathcal{L}$ is *NM*-entailed by $\mathcal{H}$, denoted by $\mathcal{H} \models_{NM} \varphi$ iff, for each *NM*-model $\mathcal{I}$ of $\mathcal{H}$, $\mathcal{I}$ satisfies φ.

In other words, the nonmonotonic semantics for a safe hybrid KB $\mathcal{H} = (\mathcal{T},\mathcal{P})$ is obtained in the following way. Take a first-order interpretation $\mathcal{I} = \mathcal{I}_P \cup \mathcal{I}_R$ such that $\mathcal{I}_P$ satisfies $\mathcal{T}$; then, evaluate $\mathcal{P}$ in $\mathcal{I}_P$, obtaining the program $\Pi(gr(\mathcal{P},\mathcal{C}),\mathcal{I}_P)$; if $\mathcal{I}_R$ represents a stable model for such a program, then $\mathcal{I}$ is a *NM*-model for $\mathcal{H}$.

It can be shown that satisfiability of safe hybrid KBs under the first-order semantics can be reduced to satisfiability under the nonmonotonic semantics (due to space limits, we are not able to provide details about this aspect in the paper). Therefore, in the rest of the paper, we study safe hybrid KBs under the nonmonotonic semantics. In particular, when we speak about satisfiability of safe hybrid KBs we always mean satisfiability under the nonmonotonic semantics.

OWA vs. CWA We now briefly comment on how the OWA of the structural part and the CWA of the relational part coexist in safe hybrid KBs.

The key point is the fact that, in safe hybrid KBs, structural predicates and relational predicates are interpreted in a different way. More precisely, the semantics of the relational part is defined starting from a given interpretation of the structural component: given an interpretation $\mathcal{I}$ of $\mathcal{T}$, we compute the stable models of the projection of $\mathcal{P}_g$ with respect to $\mathcal{I}$. In this way, it is possible to interpret relational predicates under a CWA (actually, the stable model semantics), while keeping the interpretation of structural predicates open, i.e., based on the classical FOL semantics.

Example 1. Let $\mathcal{H}$ be the safe hybrid KB where the following structural component $\mathcal{T}$ defines an ontology about persons:

$\forall x.\mathit{PERSON}(x) \rightarrow \exists y.\mathit{FATHER}(y,x) \wedge \mathit{MALE}(y)$
$\forall x.\mathit{MALE}(x) \rightarrow \mathit{PERSON}(x)$
$\forall x.\mathit{FEMALE}(x) \rightarrow \mathit{PERSON}(x)$
$\forall x.\mathit{FEMALE}(x) \rightarrow \neg\mathit{MALE}(x)$
$\mathit{MALE}(\mathit{Bob})$
$\mathit{PERSON}(\mathit{Mary})$
$\mathit{PERSON}(\mathit{Paul})$

and the rule component $\mathcal{P}$ defines nonmonotonic rules about students, as follows:

$\mathit{boy}(X) \leftarrow \mathit{enrolled}(X,c1), \mathit{PERSON}(X), \mathit{not}\ \mathit{girl}(X)$ $[R1]$
$\mathit{girl}(X) \leftarrow \mathit{enrolled}(X,c2), \mathit{PERSON}(X)$ $[R2]$
$\mathit{boy}(X) \vee \mathit{girl}(X) \leftarrow \mathit{enrolled}(X,c3), \mathit{PERSON}(X)$ $[R3]$
$\mathit{FEMALE}(X) \leftarrow \mathit{girl}(X)$ $[R4]$
$\mathit{MALE}(X) \leftarrow \mathit{boy}(X)$ $[R5]$
$\mathit{enrolled}(\mathit{Paul},c1)$
$\mathit{enrolled}(\mathit{Mary},c1)$

enrolled$(Mary, c2)$
enrolled$(Bob, c3)$

It can be easily verified that all *NM*-models for $\mathcal{H}$ satisfy the following ground atoms:

- *boy*(*Paul*) (since rule R1 is always applicable for $X = Paul$ and R1 acts like a *default rule*, which can be read as follows: if X is a person enrolled in course $c1$, then X is a boy, unless we know for sure that X is a girl)
- *girl*(*Mary*) (since rule R2 is always applicable for $X = Mary$)
- *boy*(*Bob*) (since rule R3 is always applicable for $X = Bob$, and, by rule R4, the conclusion *girl*(*Bob*) is inconsistent with $\mathcal{T}$)
- *MALE*(*Paul*) (due to rule R5)
- *FEMALE*(*Mary*) (due to rule R4)

Notice that $\mathcal{H} \models_{NM}$ *FEMALE*(*Mary*), while $\mathcal{T} \not\models_{FOL}$ *FEMALE*(*Mary*). In other words, adding a rule component has indeed an effect on the conclusions one can draw about structural predicates. Such an effect also holds under the first-order semantics of safe hybrid KBS, since it can be immediately verified that in this case $\mathcal{H} \models_{FOL}$ *FEMALE*(*Mary*). □

Among other things, the above example shows that, in safe hybrid KBs, the information flow is bidirectional: not only the structural component constrains the forms of the stable models of the rule component (through the structural predicates in the body of the rules), but also vice versa, since the rule component imposes constraints that the models of the structural components must satisfy. Hence, the rule component has an effect on the conclusions that can be drawn from the structural component, since it filters out those models $\mathcal{I}$ of the structural component for which the program $\Pi(gr(\mathcal{P}, \mathcal{C}), \mathcal{I})$ has no stable models.

UNA vs. non-UNA The semantic issue concerning the UNA is treated in safe hybrid KBs in the following way:

- The equality predicate is a structural predicate, therefore its semantics is "under control" of the structural KB, and is interpreted under the classical FOL semantics. In particular, equality is not involved in the computation of stable models, since stable models of the relational part are defined based only on a particular interpretation of the equality predicate;
- Nevertheless, new equalities may be imposed by the relational component (just like any other structural predicate), since rules may have equality atoms in the head.

Example 2. Let $\mathcal{H} = (\mathcal{T}, \mathcal{P})$ where $\mathcal{P}$ is the program constituted by the fact $r(a, b)$ and the rule *equal*$(X, Y) \leftarrow r(X, Y)$, and suppose $\mathcal{T} \cup \{$*equal*$(a, b)\}$ is satisfiable. Then, $\mathcal{H}$ is satisfiable, and *equal*(a, b) holds in every model for $\mathcal{H}$. Indeed, the effect of the relational component is to eliminate from the set of models of $\mathcal{H}$ all the interpretations $\mathcal{I}$ of the structural predicates in which $a^{\mathcal{I}} \neq b^{\mathcal{I}}$, since for such interpretations the projection of $\mathcal{P}_g$ with respect to $\mathcal{I}$ is a program that has no stable models. □

3 Reasoning in safe hybrid KBs

We now study satisfiability in safe hybrid KBs, i.e., the basic reasoning task in this framework (entailment can be easily reduced to unsatisfiability). We first define an algorithm for deciding satisfiability of safe hybrid KBs, and prove its correctness; Then, based on such an algorithm, we analyze decidability and complexity of reasoning in safe hybrid KBs; Finally, we prove decidability of OWL-DL with DL-safe rules.

Algorithm We start by providing some preliminary definitions. First, we introduce the notion of *rectification* of a Datalog$^{\neg\vee}$ program [6], which will be needed in the algorithm to properly handle the effects of the non-UNA-based semantics of the structural component on the relational component.

Definition 2. *Let R be a Datalog$^{\neg\vee}$ rule. We denote by rectify(R) the Datalog$^{\neg\vee}$ rule obtained from R as follows:*

1. *for each variable X which occurs $n \geq 2$ times in R, and for each $i \in \{1, \ldots, n\}$, replace the i-th occurrence in R of the variable X with the new variable symbol X^i;*
2. *for each variable X which occurs $n \geq 2$ times in R, and for each $i \in \{2, \ldots, n\}$, add the atom equal(X^{i-1}, X^i) to the body of the rule.*
3. *for each constant c occurring in R and not occurring within the predicate equal, replace every occurrence of c with the new variable symbol X^c, and add the atom equal(X^c, c) to the body of the rule.*

Given a Datalog$^{\neg\vee}$ program $\mathcal{P}$, we denote by rectify$(\mathcal{P})$ the program rectify$(\mathcal{P}) = \bigcup_{R\in\mathcal{P}}$ rectify(R).

Then, we introduce a notion of grounding of a relational component of a safe hybrid KB. Given a Datalog$^{\neg\vee}$ program $\mathcal{P}$, we denote by $\mathcal{C}_\mathcal{P}$ the set of constant symbols occurring in $\mathcal{P}$, and denote by $\mathcal{A}_P/\mathcal{P}$ the set of predicates from $\mathcal{A}_P$ occurring in $\mathcal{P}$. We assume that $\mathcal{A}_P$ always contains the equality predicate, even if such a predicate does not actually occur in $\mathcal{P}$.

Definition 3. *Let $\mathcal{H} = (\mathcal{T}, \mathcal{P})$ be a safe hybrid KB. The* grounding of the structural predicates in $\mathcal{P}$*, denoted by $gr_p(\mathcal{P})$ is the set of ground atoms*

$$\{m(t) \mid m \in \mathcal{A}_P/\mathcal{P} \text{ and } m \text{ has arity } k \text{ and } t \text{ is a } k\text{-tuple of constants of } \mathcal{C}_\mathcal{P}\}$$

The idea behind the above definition is that, in the case of safe hybrid KBs, $gr_p(\mathcal{P})$ identifies the set of *all* the relevant instantiations of the predicates in $\mathcal{A}_P$ needed to decide satisfiability of the rule component of the safe hybrid KB $\mathcal{H}$: In fact, due to the safeness condition in the program rules, it turns out that we can restrict the grounding of the rules only to the instantiations which substitute each variable with a symbol in $\mathcal{C}_\mathcal{P}$ (notice that, since we assume that $\mathcal{A}_P/\mathcal{P}$ always contains the equality predicate, $gr_p(\mathcal{P})$ always contains all the atoms representing the equality between two constants in $\mathcal{C}_\mathcal{P}$).

Algorithm Safe-Hybrid-Sat($\mathcal{H}$)
Input: safe hybrid KB $\mathcal{H} = (\mathcal{T}, \mathcal{P})$
Output: true if $\mathcal{H}$ is satisfiable, false otherwise
begin
  if there exists partition (G_P, G_N) of $gr_p(\mathcal{P})$
  such that
    (a) (G_P, G_N) is consistent with $\mathcal{T}$ **and**
    (b) *rectify*$(\mathcal{P}(G_P, G_N))$ has a standard stable model
  then return true
  else return false
end

Fig. 1. The algorithm Safe-Hybrid-Sat

Thus, we can divide the set of all interpretations for $\mathcal{T}$ into equivalence classes, based on the way in which such interpretations evaluate the ground atoms in $gr_p(\mathcal{P})$. Each such equivalence class can be represented by a partition (G_P, G_N) of $gr_p(\mathcal{P})$. More precisely, G_P is the set of ground atoms in $gr_p(\mathcal{P})$ satisfied by the interpretations in the equivalence class, while G_N is the set of atoms in $gr_p(\mathcal{P})$ which are not satisfied by such interpretations.

However, not all the partitions of $gr_p(\mathcal{P})$ represent a guess of the ground atoms that is compatible with the KB $\mathcal{T}$. The following definition formalizes the notion of consistency of a partition of ground atoms with respect to $\mathcal{T}$.

Definition 4. *A partition (G_P, G_N) of $gr_p(\mathcal{P})$* is consistent with $\mathcal{T}$ *iff the first-order theory $\mathcal{T} \cup \{m(t) \mid m(t) \in G_P\} \cup \{\neg m(t) \mid m(t) \in G_N\}$ is satisfiable.*

Informally, the above definition indicates that, if a partition is consistent with $\mathcal{T}$, then there exists at least one interpretation that both satisfies $\mathcal{T}$ and evaluates the atoms in $gr_p(\mathcal{P})$ according to the partition (G_P, G_N).

Finally, we denote by $\mathcal{P}(G_P, G_N)$ the Datalog$^{\neg\vee}$ program

$$\mathcal{P}(G_P, G_N) = \mathcal{P} \cup G_P \cup \{\leftarrow r(t) \mid r(t) \in G_N\}$$

In Figure 1 we report the algorithm Safe-Hybrid-Sat for deciding satisfiability of a safe hybrid KB $\mathcal{H} = (\mathcal{T}, \mathcal{P})$. The algorithm formalizes the idea that a way to decide satisfiability of $\mathcal{H}$ is to look for a partition of $gr_p(\mathcal{P})$ that is consistent with $\mathcal{T}$ and such that the program *rectify*$(\mathcal{P}(G_P, G_N))$ has a *standard* stable model, i.e., a stable model according to the standard, UNA-based semantics of Datalog$^{\neg\vee}$ [7].

More precisely, we reduce reasoning in the absence of UNA to reasoning in the presence of UNA in the relational component as follows:

- a partition (G_P, G_N) of $gr_p(\mathcal{P})$ fixes an interpretation of the equality predicate for the constants in $\mathcal{C}_\mathcal{P}$ (since all ground atoms stating equality between constants in $\mathcal{C}_\mathcal{P}$ belong to $gr_p(\mathcal{P})$);

– now, the program $\mathcal{P}(G_P, G_N)$ takes into account such an interpretation of equality by adding the corresponding facts and constraints to $\mathcal{P}$. However, to correctly model the absence of the UNA, each rule must be transformed (rectified) as in Definition 2. In fact, it can be shown [6] that the transformation of a rule produced by the rectification precisely corresponds to allow for unification of terms via the equality predicate under UNA, thus simulating the absence of the UNA in the actual semantics for safe hybrid KBs.

Example 3. Let $\mathcal{H} = (\mathcal{T}, \mathcal{P})$ where $\mathcal{P}$ is the following program:

$$\begin{aligned} \mathit{equal}(X, Y) &\leftarrow r(X, Y), r(X, Z) \\ t(X) &\leftarrow s(X, X) \\ r(a, b)& \\ r(a, c)& \\ s(b, c)& \end{aligned}$$

and for simplicity suppose that $\mathcal{T}$ is the empty theory. Let G_P, G_N be as follows:

$$\begin{aligned} G_P &= \{\mathit{equal}(b, c), \mathit{equal}(c, b), \mathit{equal}(a, a), \mathit{equal}(b, b), \mathit{equal}(c, c)\} \\ G_N &= \{\mathit{equal}(a, c), \mathit{equal}(c, a), \mathit{equal}(a, b), \mathit{equal}(b, a)\} \end{aligned}$$

First, (G_P, G_N) is consistent with $\mathcal{T}$, since it does not violate the semantics of *equal* (i.e., the fact that *equal* is an equivalence relation). Then, *rectify*$(\mathcal{P}(G_P, G_N))$ is the following program:

$$\begin{aligned} \mathit{equal}(X^1, Y^1) &\leftarrow r(X^2, Y^2), r(X^3, Z), \mathit{equal}(X^1, X^2), \mathit{equal}(X^2, X^3), \mathit{equal}(Y^1, Y^2) \\ t(X^1) &\leftarrow s(X^2, X^3), \mathit{equal}(X^1, X^2), \mathit{equal}(X^2, X^3) \\ r(X^a, X^b) &\leftarrow \mathit{equal}(X^a, a), \mathit{equal}(X^b, b) \\ r(X^a, X^c) &\leftarrow \mathit{equal}(X^a, a), \mathit{equal}(X^c, c) \\ s(X^b, X^c) &\leftarrow \mathit{equal}(X^b, b), \mathit{equal}(X^c, c) \\ &\leftarrow \mathit{equal}(a, c) \\ &\leftarrow \mathit{equal}(c, a) \\ &\leftarrow \mathit{equal}(a, b) \\ &\leftarrow \mathit{equal}(b, a) \\ \mathit{equal}(b, c)& \\ \mathit{equal}(c, b)& \\ \mathit{equal}(a, a)& \\ \mathit{equal}(b, b)& \\ \mathit{equal}(c, c)& \end{aligned}$$

It is immediate to verify that, for instance, the facts $s(b, b)$, $t(b)$, $s(c, c)$, $t(c)$ belong to the only standard stable model of *rectify*$(\mathcal{P}(G_P, G_N))$. It is also easy to see that the only other guess (G_P, G_N) that is both satisfiable at step (a) and at step (b) of the algorithm is the one in which $G_N = \emptyset$, i.e., the three constants are assumed as equal. In fact, every other guess is either unsatisfiable at step (a) of the algorithm (since it violates the fact that *equal* must be an equivalence relation) or is such that there are no stable models for *rectify*$(\mathcal{P}(G_P, G_N))$ (since $\mathit{equal}(b, c) \in G_N$ and therefore the first rule of the program is violated). $\Box$

The algorithm Safe-Hybrid-Sat is sound and complete with respect to the nonmonotonic semantics defined in Section 2.2, as stated by the following theorem.

Theorem 1. *Let $\mathcal{H} = (\mathcal{T}, \mathcal{P})$ be a safe hybrid KB. Then, $\mathcal{H}$ is satisfiable iff Safe-Hybrid-Sat($\mathcal{H}$) returns* true.

We remark that the algorithm reduces reasoning in safe hybrid KBs to standard reasoning in the structural component (step (a)) and to standard reasoning in Datalog$^{\neg\vee}$ (step (b)). Therefore, not only the algorithm is modular, but also it allows for reusing deductive techniques (and implemented systems) developed for the structural language and for Datalog$^{\neg\vee}$ [8].

Decidability and complexity We now study decidability and complexity issues in the framework of safe hybrid KBs. We start by recalling a decidability and complexity result for Datalog$^{\neg\vee}$ programs under standard (UNA-based) stable model semantics.

Proposition 1 ([7]). *Satisfiability of Datalog$^{\neg\vee}$ programs under standard stable model semantics is NEXPTIME^{NP}-complete. Moreover, satisfiability of Datalog$^{\neg}$ programs under standard stable model semantics is NEXPTIME-complete.*

Then, it can be shown that satisfiability of Datalog$^{\neg\vee}$ programs under standard stable model semantics can be reduced to satisfiability in safe hybrid KBs. Consequently, the following hardness result follows.

Theorem 2. *Satisfiability of safe hybrid KBs is NEXPTIME^{NP}-hard. Moreover, it is NEXPTIME-hard if the rule component is a Datalog$^{\neg}$ program.*

We now prove a very general result on the decidability of reasoning in safe hybrid KBs.

Theorem 3. *Let $\mathcal{H} = (\mathcal{T}, \mathcal{P})$ be a safe hybrid KB. If establishing consistency of a partition of $gr_p(\mathcal{P})$ with $\mathcal{T}$ is decidable, then satisfiability of $\mathcal{H}$ is a decidable problem.*

Proof. First, observe that the set $gr_p(\mathcal{P})$ is finite, therefore the number of partitions of $gr_p(\mathcal{P})$ is finite. Then, since by hypothesis establishing consistency of a partition (G_P, G_N) with $\mathcal{T}$ is decidable, for each such partition (G_P, G_N) condition (a) of the algorithm can be verified in a finite amount of time; moreover, since *rectify*$(\mathcal{P}(G_P, G_N))$ is a finite Datalog$^{\neg\vee}$ program, from Proposition 1 it follows that condition (b) of the algorithm can also be verified in a finite amount of time. □

We remark that, starting from a logic $\mathcal{L}$ in which reasoning is decidable, it is very often the case that deciding satisfiability of a theory of an $\mathcal{L}$-KB augmented with a finite set of ground literals is still decidable, and therefore that reasoning in safe hybrid KBs made of $\mathcal{L}$ theories as structural components is decidable. In this sense, the previous theorem can be read as a very strong result, stating that the framework of safe hybrid KBs generally preserves decidability of reasoning.

Decidability of OWL-DL with DL-safe rules The DL that currently plays a central role in the Semantic Web is $\mathcal{SHOIN}(\mathbf{D})$: as mentioned in Section 1, it is equivalent to OWL-DL [19], which is a W3C recommendation language for ontology representation in the Semantic Web. Reasoning in $\mathcal{SHOIN}(\mathbf{D})$, and hence in OWL-DL, is decidable, as stated by the following property.

Proposition 2 ([14, 22]). *Satisfiability of $\mathcal{SHOIN}(\mathbf{D})$ KBs is NEXPTIME-complete.*

Based on Theorem 3, it is possible to prove that reasoning in $\mathcal{SHOIN}(\mathbf{D})$ safe hybrid KBs is decidable, and to provide a computational characterization of the problem.

Theorem 4. *Let $\mathcal{H} = (\mathcal{T}, \mathcal{P})$ be a safe hybrid KB where $\mathcal{T}$ is a $\mathcal{SHOIN}(\mathbf{D})$ KB and $\mathcal{P}$ is a Datalog$^{\neg\vee}$ program. Deciding satisfiability of $\mathcal{H}$ is NEXPTIME^{NP}-complete. Moreover, if $\mathcal{P}$ is a Datalog$^{\neg}$ program, deciding satisfiability of $\mathcal{H}$ is NEXPTIME-complete.*

As a corollary of the above theorem, we close an open problem in [18], i.e., decidability of satisfiability of $\mathcal{SHOIN}(\mathbf{D})$ with DL-safe rules. This problem exactly corresponds in our framework to deciding satisfiability of a safe hybrid KB composed of a $\mathcal{SHOIN}(\mathbf{D})$ KB and a positive Datalog program: as a corollary of the above results, it immediately follows that satisfiability in such safe hybrid KBs is decidable and is NEXPTIME-complete.

4 Related work

Although in various forms, the notion of safe integration has been taken into account since the earliest studies concerning the extension of DLs with rules. The first formal proposal for the integration of Description Logics and rules is $\mathcal{AL}$-log [5]. $\mathcal{AL}$-log is a framework which integrates KBs expressed in the description logic $\mathcal{ALC}$ and positive Datalog programs. Then, disjunctive $\mathcal{AL}$-log was proposed in [20] as an extension of $\mathcal{AL}$-log, based on the use of Datalog$^{\neg\vee}$ instead of positive Datalog, and on the possibility of using binary predicates (roles) besides unary predicates (concepts) in rules. When choosing $\mathcal{ALC}$ as the structural language, the framework of safe hybrid KBs captures disjunctive $\mathcal{AL}$-log and can be seen as a generalization of it: indeed, differently from safe hybrid KBs, in disjunctive $\mathcal{AL}$-log structural predicates can occur only in the bodies of rules, which restricts the information flow only from the structural KB to the rule KB, but not vice versa.

This line of research was carried on by the work on CARIN [16], which established several fundamental decidability results concerning non-safe interaction between DL-KBs and rules. Some of such results clearly indicate that, in case of unrestricted interaction between the structural component and the rule component in hybrid KBs, decidability of reasoning holds only if at least one of the two component KBs has very limited expressive power: e.g., in order to retain decidability of reasoning, allowing recursion in the rule KB imposes very severe restrictions on the expressiveness of the structural KB.

The framework of $\mathcal{AL}$-log has been extended in a different way in [18]. There, the problem of extending OWL-DL with positive Datalog programs is analyzed. The interaction between OWL-DL and rules is restricted through a safeness condition which is exactly the one adopted in safe hybrid KBs. With respect to disjunctive $\mathcal{AL}$-log, in [18] a more expressive structural language and a less expressive rule language are adopted: moreover, the information flow is bidirectional, i.e., structural predicates may appear in the head of rules. As we have shown in Section 3, such a framework is perfectly captured by safe hybrid KBs.

The work presented in [11] can also be seen as an approach based on a form of safe interaction between the structural DL-KB and the rules: in particular, a rule language is defined such that it is possible to encode a set of rules into a semantically equivalent DL-KB. As a consequence, such a rule language is very restricted.

A different approach is presented in [13, 12], which proposes Conceptual Logic Programming (CLP), an extension of answer set programming (i.e., Datalog$^{\neg\vee}$) towards infinite domains. In order to keep reasoning decidable, a syntactic restriction on CLP program rules is imposed. This approach is related to integrating DLs and rules, since the authors also show that CLPs can embed expressive DL-KBs, which in turn implies decidability of adding CLP rules to such DLs. However, the syntactic restriction on CLP rules, whose purpose is to impose a "forest-like" structure to the models of the program, is different from the safeness conditions analyzed so far, which makes it impossible to compare this approach with safe hybrid KBs (and with the approaches previously mentioned).

Another approach for extending DLs with Datalog$^{\neg}$ rules is presented in [9]. Differently from safe hybrid KBs and from the other approaches above described, this proposal allows for specifying in rule bodies *queries* to the structural component, where every query also allows for specifying an input from the rule component, and thus for an information flow from the rule component to the structural component. The meaning of such queries in rule bodies is given at the meta-level, through the notion of skeptical entailment in the DL-KB. Thus, from the semantic viewpoint, this form of interaction-via-entailment between the two components is more restricted than in safe hybrid KBs (and in the similar approaches previously mentioned); on the other hand, such an increased separation in principle allows for more modular reasoning methods, which are able to completely separate reasoning about the structural component and reasoning about the rule component. However, in this paper we have shown that an analogous form of modularization of reasoning is possible also in the presence of a semantically richer form of interaction between the two components of a safe hybrid KB.

An approach for the combination of defeasible reasoning with Description Logics is presented in [1], under a safe interaction-via-entailment scheme which is semantically analogous to the one proposed in [9]. Besides the differences with our approach (and with the studies on nonmonotonic extensions of DL-KBs previously mentioned) concerning the semantics of nonmonotonic rules, a main characteristic of these proposals consists in the fact the information flow is unidirectional, i.e., it goes from the structural component to the rule component.

Generally speaking, it is difficult to provide a satisfactory semantic account for non-safe interaction between DL-KBs and nonmonotonic rules, due to the classical, open world semantics of DL-KBs, and the closed world assumption underlying non-monotonic systems. For instance, [17] illustrates the problems in providing a semantic account for non-safe interaction of ontologies and Datalog$^{\neg\vee}$ programs.

Finally, [4] proposes OWL Flight, a logic programming based formalism for the Semantic Web. A detailed comparison of the relative expressive abilities of OWL Flight and OWL-DL is made, which proves the adequacy of the proposed approach for Semantic Web applications. Although based on logic program rules, the purpose of this approach is different from ours and from the ones mentioned above, and does not actually deal with the problem of integrating DLs with rules.

5 Conclusions

In this paper we have formally demonstrated that the form of safe interaction introduced in [5] and extended in various forms by [20, 18] can be generally applied, and constitutes a good choice for the design of integrated KBs when we want to keep expressive power both in the structural component and in the rule component, and when decidability and complexity of (sound and complete) reasoning is a crucial aspect. Indeed, in general, such safe interaction preserves decidability of reasoning and, in many cases, does not increase the complexity of reasoning, i.e., reasoning in the integrated KB is computationally no harder than reasoning separately in the two components.

Moreover, we have shown that such a form of safe interaction allows for a clear formal treatment of hybrid KBs in which the UNA is not adopted, and in which we want the OWA on the structural component and the CWA on the rule component.

A possible further extension of the present work is towards the study of *data complexity* in the framework of safe hybrid KBs, following the lines of [3], which analyzes data complexity for $\mathcal{AL}$-log. Moreover, it should be interesting to analyze whether tighter forms of interaction between the structural and the rule component can be defined, relaxing, on the one hand, the safeness condition of safe hybrid KBs, while preserving, on the other hand, their nice computational properties. Finally, it would be very interesting to study *data* complexity in the framework of safe hybrid KBs, continuing the research presented in [3], which analyzes data complexity for $\mathcal{AL}$-log.

Acknowledgments This research has been partially supported by the projects INFOMIX (IST-2001-33570) and INTEROP Network of Excellence (IST-508011) funded by the EU, and by the project HYPER, funded by IBM through a Shared University Research (SUR) Award grant.

References

1. Grigoris Antoniou. A nonmonotonic rule system using ontologies. In *Proc. of RuleML 2002*, volume 60 of *CEUR Workshop Proceedings*, 2002.
2. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

3. Marco Cadoli, Luigi Palopoli, and Maurizio Lenzerini. Datalog and description logics: Expressive power. In *Proc. of DBPL'97*, 1997.
4. Jos de Bruijn, Ruben Lara, Axel Polleres, and Dieter Fensel. OWL DL vs. OWL flight: conceptual modeling and reasoning for the semantic web. In *Proc. of WWW 2005*, pages 623–632, 2005.
5. Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. $\mathcal{AL}$-log: Integrating Datalog and description logics. *J. of Intelligent Information Systems*, 10(3):227–252, 1998.
6. Oliver M. Duschka, Michael R. Genesereth, and Alon Y. Levy. Recursive query plans for data integration. *J. of Logic Programming*, 43(1):49–73, 2000.
7. Thomas Eiter, Georg Gottlob, and Heikki Mannilla. Disjunctive Datalog. *ACM Trans. on Database Systems*, 22(3):364–418, 1997.
8. Thomas Eiter, Nicola Leone, Cristinel Mateis, Gerald Pfeifer, and Francesco Scarcello. The KR system dlv: Progress report, comparison and benchmarks. In *Proc. of KR'98*, pages 636–647, 1998.
9. Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits. Combining answer set programming with description logics for the semantic web. In *Proc. of KR 2004*, pages 141–151, 2004.
10. Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
11. Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: combining logic programs with description logic. In *Proc. of WWW 2003*, pages 48–57, 2003.
12. Stijn Heymans, Davy Van Nieuwenborgh, and Dirk Vermeir. Semantic web reasoning with conceptual logic programs. In *Proc. of RuleML 2004*, pages 113–127, 2004.
13. Stijn Heymans and Dirk Vermeir. Integrating description logics and answer set programming. In *Proc. of PPSWR 2003*, pages 146–159, 2003.
14. Ian Horrocks and Peter F. Patel-Schneider. Reducing OWL entailment to Description Logic satisfiability. In *Proc. of ISWC 2003*, pages 17–29, 2003.
15. Ian Horrocks and Peter F. Patel-Schneider. A proposal for an OWL rules language. In *Proc. of WWW 2004*, pages 723–731, 2004.
16. Alon Y. Levy and Marie-Christine Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104(1–2):165–209, 1998.
17. Jing Mei, Shengping Liu, Anbu Yue, and Zuoquan Lin. An extension to OWL with general rules. In *Proc. of RuleML 2004*, pages 155–169, 2004.
18. Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. In *Proc. of ISWC 2004*, pages 549–563, 2004.
19. Peter F. Patel-Schneider, Patrick J. Hayes, Ian Horrocks, and Frank van Harmelen. OWL web ontology language; semantics and abstract syntax. W3C candidate recommendation, http://www.w3.org/tr/owl-semantics/, november 2002.
20. Riccardo Rosati. Towards expressive KR systems integrating Datalog and description logics: Preliminary report. In *Proc. of DL'99*, pages 160–164. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-22/`, 1999.
21. Riccardo Rosati. On the decidability and complexity of integrating ontologies and rules. *Journal of Web Semantics*, 2005. To appear.
22. Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.