$\mathcal{DL}$+*log*: Tight Integration of Description Logics and Disjunctive Datalog

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

Abstract

The integration of Description Logics and Datalog rules presents many semantic and computational problems. In particular, reasoning in a system fully integrating Description Logics knowledge bases (DL-KBs) and Datalog programs is undecidable. Many proposals have overcomed this problem through a "safeness" condition that limits the interaction between the DL-KB and the Datalog rules. Such a safe integration of Description Logics and Datalog provides for systems with decidable reasoning, at the price of a strong limitation in terms of expressive power. In this paper we define $\mathcal{DL}$+*log*, a general framework for the integration of Description Logics and disjunctive Datalog. From the knowledge representation viewpoint, $\mathcal{DL}$+*log* extends previous proposals, since it allows for a tighter form of integration between DL-KBs and Datalog rules which overcomes the main representational limits of the approaches based on the safeness condition. From the reasoning viewpoint, we present algorithms for reasoning in $\mathcal{DL}$+*log*, and prove decidability and complexity of reasoning in $\mathcal{DL}$+*log* for several Description Logics. To the best of our knowledge, $\mathcal{DL}$+*log* constitutes the most powerful decidable combination of Description Logics and disjunctive Datalog rules proposed so far.

Introduction

The problem of adding rules to Description Logics is currently a hot research topic, due to the interest of Semantic Web applications towards the integration of rule-based systems with ontologies (Horrocks & Patel-Schneider 2004). Practically all the approaches in this field concern the study of description logic knowledge bases (DL-KBs) augmented with rules expressed in Datalog and its nonmonotonic extensions.

Many technical problems arise in this kind of KR systems. In particular, the full interaction between a DL-KB and a Datalog program easily leads to undecidability of reasoning (Levy & Rousset 1998) and to semantic problems related to the simultaneous presence of knowledge interpreted under a classical open-world assumption (the DL-KB) and knowledge interpreted under a closed-world assumption (nonmonotonic Datalog rules).

Several proposals in the field (e.g., (Donini *et al.* 1998; Motik, Sattler, & Studer 2004; Eiter *et al.* 2004a; Rosati 2005a)) are based on the idea of solving the above problems by restricting the interaction between DL-KBs and Datalog rules through a *DL-safeness* condition, which restricts the use of variables in Datalog rules. Informally, DL-safeness imposes that each variable in a Datalog rule occurs in special predicates which cannot occur as any other predicate (concept or role) in the DL-KB, in a way such that the variables are bound to range only over the constants explicitly mentioned in the DL-KB. This technical restriction actually allows for overcoming both the undecidability and the semantic problems mentioned above (for a detailed discussion on this topic, see e.g. (Motik, Sattler, & Studer 2005; Rosati 2005b)).

However, the DL-safeness condition imposes a severe restriction on the expressiveness of the overall KR system: e.g., DL-safe rules are not able to express arbitrary *conjunctive queries* to the DL-KB. Conjunctive queries correspond to a simple form of non-recursive Datalog rules, are computable in many DLs and there are known algorithms for conjunctive query algorithms in many DLs (Calvanese, De Giacomo, & Lenzerini 1998; Ortiz de la Fuente *et al.* 2005). Therefore, DL-safeness seems to imply an unnecessary limitation in the expressiveness of rules.

In this paper we try to overcome the limitations of the DL-safe integration of DLs and Datalog, and present $\mathcal{DL}$+*log*, a general framework for the integration of Description Logics and disjunctive Datalog (Eiter, Gottlob, & Mannilla 1997). With respect to DL-safe based approaches, $\mathcal{DL}$+*log* realizes a tighter form of interaction between DL-KBs and rules, through a new safeness condition (*weak safeness*) that weakens DL-safeness of variables in Datalog rules.

Such a tighter integration allows for an increase in the expressive power: conjunctive queries (and unions of conjunctive queries) can be actually expressed in $\mathcal{DL}$+*log* through weakly-safe rules, thus overcoming the main representational limits of the approaches based on the DL-safeness condition.

At the same time, we prove that the weakly-safe interaction between DL-KBs and rules is still decidable in many DLs, by exploiting the deep relationship between query containment in DLs and reasoning in $\mathcal{DL}$+*log*. More precisely, we show the correspondence between satisfiability in $\mathcal{DL}$+*log* and containment between a conjunctive query and a union of conjunctive queries in DLs. Based on such a corre-

spondence, we provide algorithms for reasoning in $\mathcal{DL}$+*log*.

To the best of our knowledge, $\mathcal{DL}$+*log* constitutes the most powerful decidable combination of Description Logics and disjunctive (nonmonotonic) Datalog rules proposed so far, and one of the most powerful approaches among the decidable combinations of DLs and positive recursive Datalog rules. The only approach in this last class that we know of and that is not subsumed by $\mathcal{DL}$+*log* is role-safe recursive CARIN (Levy & Rousset 1998), which is uncomparable with $\mathcal{DL}$+*log*.

The paper is structured as follows. In the next section, we define the framework of $\mathcal{DL}$+*log* for the integration of DLs and rules. Then, we study reasoning in $\mathcal{DL}$+*log*, and define an algorithm for satisfiability in $\mathcal{DL}$+*log*. In the subsequent section, we address decidability and complexity of reasoning in $\mathcal{DL}$+*log*. Finally, we briefly discuss related work and draw some conclusions. Due to space limits, proofs of theorems are omitted in the present version of the paper.

$\mathcal{DL}$+*log*

The framework of $\mathcal{DL}$+*log*, i.e., $\mathcal{DL}$-KBs with weakly-safe disjunctive Datalog (Datalog$^{\neg\vee}$) rules, that we introduce in this section, constitutes an extension of $\mathcal{DL}$-*log*, originally proposed in (Rosati 1999) and then extended to the framework of r-hybrid KBs presented in (Rosati 2005a; 2005b). We thus refer to (Rosati 2005a) for more details about the general framework. In the following, we assume that the reader is familiar with the basics of Description Logics (Baader *et al.* 2003).

Syntax

We start from three mutually disjoint predicate alphabets:

- an alphabet of concept names Σ_C;
- an alphabet of role names Σ_R;
- an alphabet of *Datalog predicates* Σ_D.

We call a predicate p a *DL-predicate* if either $p \in \Sigma_C$ or $p \in \Sigma_R$. Then, we denote by $\mathcal{C}$ a countably infinite alphabet of constant names.

An *atom* is an expression of the form $p(X)$, where p is a predicate of arity n and X is a n-tuple of variables and constants. If no variable symbol occurs in X, then $p(X)$ is called a *ground atom* (or *fact*). If $p \in \Sigma_C \cup \Sigma_R$, the atom is called a *DL-atom*, while if $p \in \Sigma_D$, it is called a *Datalog atom*.

We recall (see (Eiter, Gottlob, & Mannilla 1997)) that a Datalog$^{\neg\vee}$ rule R is an expression of the form

$$p_1(X_1) \vee \ldots \vee p_n(X_n) \leftarrow r_1(Y_1), \ldots, r_m(Y_m), not\, u_1(W_1), \ldots, not\, u_h(W_h)$$

such that $n \geq 0$, $m \geq 0$, $h \geq 0$, each $p_i(X_i)$, $r_i(Y_i)$, $u_i(W_i)$ is an atom and every variable occurring in R must appear in at least one of the atoms $r_1(Y_1), \ldots, r_m(Y_m)$. This last condition is known as the *Datalog safeness* condition for variables. The variables occurring in the atoms $p_1(X_1), \ldots, p_n(X_n)$ are called the *head variables* of R. If $n = 0$, we call R a *constraint*.

A a *Datalog*$^{\neg\vee}$ program is a set of Datalog$^{\neg\vee}$ rules. If, for all $R \in \mathcal{P}$, $n \leq 1$, $\mathcal{P}$ is called a *Datalog*$^{\neg}$ program. If, for all $R \in \mathcal{P}$, $h = 0$, $\mathcal{P}$ is called a *positive disjunctive Datalog* 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.

Definition 1 *Given a description logic $\mathcal{DL}$, a $\mathcal{DL}$-*knowledge base with weakly-safe Datalog$^{\neg\vee}$ rules *($\mathcal{DL}$+log-KB for short) $\mathcal{B}$ is a pair $(\mathcal{K}, \mathcal{P})$, where:*

- *$\mathcal{K}$ is a $\mathcal{DL}$-KB, i.e., a pair $(\mathcal{T}, \mathcal{A})$ where $\mathcal{T}$ is the* TBox *and $\mathcal{A}$ is the* ABox *(Baader* et al. *2003);*
- *$\mathcal{P}$ is a set of Datalog$^{\neg\vee}$ rules, where each rule R has the form*

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

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 either a DL-predicate or a Datalog predicate;*
  - *each r_i, u_i is a Datalog predicate;*
  - *each s_i is a DL-predicate;*
  - *(Datalog safeness) every variable occurring in R must appear in at least one of the atoms $r_1(Y_1), \ldots, r_m(Y_m), s_1(Z_1), \ldots, s_k(Z_k)$;*
  - (weak safeness) *every head variable of R must appear in at least one of the atoms $r_1(Y_1), \ldots, r_m(Y_m)$.*

We remark that the above notion of weak safeness allows for the presence of variables that only occur in DL-atoms in the body of R. On the other hand, the notion of *DL-safeness* of variables adopted in previous approaches (Rosati 1999; Motik, Sattler, & Studer 2005; Rosati 2005a) can be expressed as follows: *every variable of R must appear in at least one of the atoms $r_1(Y_1), \ldots, r_m(Y_m)$*. Therefore, DL-safeness forces every variable of R to occur also in the Datalog atoms in the body of R, while weak safeness allows for the presence of variables that only occur in DL-atoms in the body of R.

Without loss of generality, in the rest of the paper we assume that in a $\mathcal{DL}$+*log*-KB $(\mathcal{K}, \mathcal{P})$ all constants occurring in $\mathcal{K}$ also occur in $\mathcal{P}$.

First-order semantics

The interpretation of constants is according to the *standard names assumption*:[1] every first-order interpretation is over the same fixed, countably infinite, domain Δ, and in addition, the alphabet of constants $\mathcal{C}$ is such that it is in the same one-to-one correspondence with Δ in every interpretation: that is, there is a constant symbol for each element of Δ, each constant denotes the same element of Δ in every interpretation, and two distinct constants denote two distinct elements (this last property is known as the *unique name assumption*).

[1] For motivation and details on the standard names assumption in this setting, see (Rosati 2005a; 2005b).

In the following, when we speak about satisfiability and query containment in DLs, we will refer to these notions under the above semantic assumption.

Let R be the following Datalog$^{\neg\vee}$ rule:

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

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

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

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

Moreover, given a $\mathcal{DL}$-KB $\mathcal{K}$, we denote by *FO*$(\mathcal{K})$ the first-order theory obtained by the standard translation of DLs into FOL (see e.g. (Baader *et al.* 2003) for details).

A *FOL-model* of a $\mathcal{DL}$*+log*-KB $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ is an interpretation $\mathcal{I}$ of $\Sigma_C \cup \Sigma_R \cup \Sigma_D$ such that $\mathcal{I}$ satisfies *FO*$(\mathcal{K}) \cup$ *FO*$(\mathcal{P})$. $\mathcal{B}$ is called *FOL-satisfiable* if it has at least a FOL-model. A ground atom $p(c)$ is *FOL-entailed* by $\mathcal{B}$ iff, for each FOL-model $\mathcal{I}$ of $\mathcal{B}$, $\mathcal{I}$ satisfies $p(c)$.

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 $A \leftarrow B, not\ C$ and $A \vee C \leftarrow B$ have the same meaning.

Nonmonotonic semantics

Given an interpretation $\mathcal{I}$ and a predicate alphabet Σ, we denote by $\mathcal{I}_\Sigma$ the projection of $\mathcal{I}$ to Σ, i.e., $\mathcal{I}_\Sigma$ is obtained from $\mathcal{I}$ by restricting it to the interpretation of the predicates in Σ.

Given a set of constants $\mathcal{C}$, 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 $\Sigma' \subset \Sigma$, and a ground program $\mathcal{P}_g$ over the predicates in Σ, 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 $r \in \Sigma'$ and $t \in r^{\mathcal{I}}$;
- delete each atom $r(t)$ in the head of R such that $r \in \Sigma'$ and $t \notin r^{\mathcal{I}}$;
- delete R if there exists an atom $r(t)$ in the body of R such that $r \in \Sigma'$ and $t \notin r^{\mathcal{I}}$;
- delete each atom $r(t)$ in the body of R such that $r \in \Sigma'$ and $t \in r^{\mathcal{I}}$;

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 $\Sigma' = \Sigma_C \cup \Sigma_R$, all occurrences of DL-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 DL-predicates occurring in $\mathcal{P}_g$.

Given two interpretations $\mathcal{I}, \mathcal{I}'$ of the set of predicates Σ, we write $\mathcal{I}' \subset_\Sigma \mathcal{I}$ if

1. for each $p \in \Sigma$, $p^{\mathcal{I}'} \subseteq p^{\mathcal{I}}$, and
2. there exists $p \in \Sigma$ such that $p^{\mathcal{I}'} \subset p^{\mathcal{I}}$.

In words, $\mathcal{I}' \subset_\Sigma \mathcal{I}$ if the extension of the predicates of Σ in $\mathcal{I}$ is strictly larger than in $\mathcal{I}'$.

Given a positive ground Datalog$^{\neg\vee}$ program $\mathcal{P}$ over an alphabet of predicates Σ 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 $\mathcal{I}' \subset_\Sigma \mathcal{I}$.

Given a ground Datalog$^{\neg\vee}$ program $\mathcal{P}$ and an interpretation $\mathcal{I}$ for $\mathcal{P}$, the *GL-reduct* (Gelfond & Lifschitz 1991) 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 $R \in \mathcal{P}$:

1. delete R if there exists a negated atom $not\ r(t)$ in the body of R such that $t \in r^{\mathcal{I}}$;
2. delete each negated atom $not\ r(t)$ in the body of R such that $t \notin r^{\mathcal{I}}$.

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*$(\mathcal{P}, \mathcal{I})$.

Definition 2 *An interpretation $\mathcal{I}$ of $\Sigma_C \cup \Sigma_R \cup \Sigma_D$ is a NM-model for $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ if the following conditions hold:*

1. *$\mathcal{I}_{\Sigma_C \cup \Sigma_R}$ satisfies $\mathcal{K}$;*
2. *$\mathcal{I}_{\Sigma_D}$ is a stable model for $\Pi(gr(\mathcal{P}, \mathcal{C}), \mathcal{I}_{\Sigma_C \cup \Sigma_R})$.*

*$\mathcal{B}$ is called NM-*satisfiable *if $\mathcal{B}$ has at least a NM-model.*

We say that a ground atom $p(c)$ is *NM*-entailed by $\mathcal{B}$ iff, for each *NM*-model $\mathcal{I}$ of $\mathcal{B}$, $\mathcal{I}$ satisfies $p(c)$.

According to the NM semantics, DL-predicates are still interpreted under the classical open-world assumption (OWA), while Datalog predicates are interpreted under a closed-world assumption (CWA) (see (Rosati 2005b) for a detailed discussion of this aspect).

Notice that, both under the FOL semantics and the NM semantics, entailment can be reduced to satisfiability, since it is possible to express constraints in the Datalog program. More precisely, under both semantics, it is immediate to verify that $(\mathcal{K}, \mathcal{P})$ entails $p(c)$ iff $(\mathcal{K}, \mathcal{P} \cup \{\leftarrow p(c)\})$ is unsatisfiable. In a similar way, it can be seen that *conjunctive query answering* can be reduced to satisfiability in $\mathcal{DL}$*+log* (see the discussion section). Consequently, in the following we concentrate on the satisfiability problem in $\mathcal{DL}$*+log*-KBs.

Relationship between FOL and NM semantics

We now show that, when the rules are positive disjunctive, i.e., there are no negated atoms in the bodies of rules, the

above two semantics are equivalent with respect to the satisfiability problem.

Theorem 3 *Let $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ be a $\mathcal{DL}$+log-KB, where $\mathcal{P}$ is a positive disjunctive Datalog program. $\mathcal{B}$ is FOL-satisfiable iff $\mathcal{B}$ is NM-satisfiable.*

Moreover, given a rule R of the form (1), we denote by $\tau(R)$ the rule obtained from R by moving every negated atom in the body of R to the rule head. Formally:

$$\begin{aligned}\tau(R) = p_1(X_1, c_1) \vee \ldots \vee p_n(X_n, c_n) \vee \\ u_1(W_1, f_1) \vee \ldots \vee u_h(W_h, f_h) \leftarrow \\ r_1(Y_1, d_1), \ldots, r_m(Y_m, d_m), \\ s_1(Z_1, e_1), \ldots, s_k(Z_k, e_k)\end{aligned}$$

Theorem 4 *Let $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ be a $\mathcal{DL}$+log-KB. $\mathcal{B}$ is FOL-satisfiable iff $\mathcal{B}' = (\mathcal{K}, \tau(\mathcal{P}))$ is NM-satisfiable, where $\tau(\mathcal{P}) = \bigcup_{R \in \mathcal{P}} \tau(R)$.*

Therefore, FOL-satisfiability can always be reduced (in linear time) to NM-satisfiability. Hence, in the following we concentrate on the satisfiability problem under the NM semantics.

We conclude this section with two simple examples of $\mathcal{DL}$+log knowledge bases. In both examples, we denote DL-predicates by uppercase names, and denote Datalog predicates by lowercase names.

Example 5 Let $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ be the $\mathcal{DL}$+log knowledge base reported in Figure 1, where the DL-KB $\mathcal{K}$ defines an ontology about persons, and the disjunctive Datalog program $\mathcal{P}$ defines nonmonotonic rules about students.

For the reader unfamiliar with the DL syntax, we report the translation of the first four inclusion assertions of the DL-KB $\mathcal{K}$ in terms of sentences in first-order logic:

$$\begin{aligned}&\forall x.\mathit{PERSON}(x) \rightarrow \exists y.\mathit{FATHER}(y, x) \wedge \mathit{MALE}(y)\\ &\forall x.\mathit{MALE}(x) \rightarrow \mathit{PERSON}(x)\\ &\forall x.\mathit{FEMALE}(x) \rightarrow \mathit{PERSON}(x)\\ &\forall x.\mathit{FEMALE}(x) \rightarrow \neg \mathit{MALE}(x)\end{aligned}$$

It can be easily verified that all *NM*-models for $\mathcal{B}$ 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 $\mathcal{K}$);
- *MALE*(*Paul*) (due to rule R5);
- *FEMALE*(*Mary*) (due to rule R4).

Notice that $\mathcal{B} \models_{NM} \mathit{FEMALE}(\mathit{Mary})$, while $\mathcal{K} \not\models_{FOL} \mathit{FEMALE}(\mathit{Mary})$. In other words, adding rules has indeed an effect on the conclusions one can draw about DL-predicates. Moreover, such an effect also holds under the first-order semantics of $\mathcal{DL}$+log-KBs, since it can be immediately verified that in this case $\mathcal{B} \models_{FOL} \mathit{FEMALE}(\mathit{Mary})$. ∎

Example 6 Let $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ be the $\mathcal{DL}$+log knowledge base reported in Figure 2.

For the reader unfamiliar with the DL syntax, we recall that the meaning of the first the first assertion of the DL-KB $\mathcal{K}$ is expressed by the first-order logic sentence

$$\begin{aligned}\forall x.\mathit{RICH}(x) \wedge \mathit{UNMARRIED}(x) \rightarrow \\ \exists y.\mathit{WANTS\text{-}TO\text{-}MARRY}(y, x)\end{aligned}$$

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

- *RICH*(*Paul*) and *RICH*(*Mary*), since the default rule R2 is always applicable for $X =$ *Paul* and $X =$ *Mary*, but not for $X =$ *Joe*, since the fact *scientist*(*Joe*) holds in every model for $\mathcal{B}$;
- $\exists \mathit{WANTS\text{-}TO\text{-}MARRY}^{-}.\top(\mathit{Mary})$, due to the first axiom of the DL-KB and to the fact that both *RICH*(*Mary*) and *UNMARRIED*(*Mary*) hold in every model of the $\mathcal{DL}$+log-KB $\mathcal{B}$ (while $\exists \mathit{WANTS\text{-}TO\text{-}MARRY}^{-}.\top(\mathit{Paul})$ is not forced by such axiom to hold in every model of $\mathcal{B}$, because *UNMARRIED*(*Paul*) is not forced to hold in every such model);
- *happy*(*Mary*), due to the above conclusions and to the rule R1. Indeed, since $\exists \mathit{WANTS\text{-}TO\text{-}MARRY}^{-}.\top(\mathit{Mary})$ holds in every model of $\mathcal{B}$, it follows that in every model there exists a constant x such that *WANTS-TO-MARRY*(x, Mary) holds in the model, consequently from rule R1 it follows that *happy*(*Mary*) also holds in the model.

Notice that, according to the definitions given in the previous section, the variable Y in rule R1 is weakly-safe but *not* DL-safe, since Y does not occur in any Datalog predicate in rule R1. ∎

Reasoning

In this section we study reasoning in $\mathcal{DL}$+log. In particular, we study satisfiability for finite $\mathcal{DL}$+log-KBs (as mentioned above, entailment can be easily reduced to satisfiability in $\mathcal{DL}$+log).

We start by introducing Boolean conjunctive queries (CQs) and Boolean unions of conjunctive queries (UCQs), and the containment problem for such queries. A Boolean UCQ over a predicate alphabet Σ is a first-order sentence of the form $\exists \vec{x}.\mathit{conj}_1(\vec{x}) \vee \ldots \vee \mathit{conj}_n(\vec{x})$, where $\vec{x}$ is a tuple of variable symbols and each $\mathit{conj}_i(\vec{x})$ is a set of atoms whose predicates are in Σ and whose arguments are either constants or variables from $\vec{x}$. A Boolean CQ corresponds to a Boolean UCQ in the case when $n = 1$.

Given a $\mathcal{DL}$-TBox $\mathcal{T}$, a Boolean CQ Q_1 and a Boolean UCQ Q_2 over the alphabet $\Sigma_C \cup \Sigma_R$, *Q_1 is contained in Q_2 with respect to $\mathcal{T}$*, denoted by $\mathcal{T} \models Q_1 \subseteq Q_2$, iff, for every model $\mathcal{I}$ of $\mathcal{T}$, if Q_1 is satisfied in $\mathcal{I}$ then Q_2 is satisfied in $\mathcal{I}$.In the following, we call the problem of deciding $\mathcal{T} \models Q_1 \subseteq Q_2$ the *Boolean CQ/UCQ containment problem*.[2]

[2]This problem was called *existential entailment* in (Levy & Rousset 1998).

$\mathit{PERSON} \sqsubseteq \exists \mathit{FATHER}^{-}.\mathit{MALE}$
$\mathit{MALE} \sqsubseteq \mathit{PERSON}$
$\mathit{FEMALE} \sqsubseteq \mathit{PERSON}$
$\mathit{FEMALE} \sqsubseteq \neg \mathit{MALE}$
$\mathit{MALE}(\mathit{Bob})$
$\mathit{PERSON}(\mathit{Mary})$
$\mathit{PERSON}(\mathit{Paul})$

(a) DL-KB $\mathcal{K}$ (ontology about persons)

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

(b) disjunctive Datalog program $\mathcal{P}$ (rules about students)

Figure 1: $\mathcal{DL}$+*log* knowledge base $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ of Example 5

$\mathit{RICH} \sqcap \mathit{UNMARRIED} \sqsubseteq \exists \mathit{WANTS\text{-}TO\text{-}MARRY}^{-}.\top$
$\mathit{UNMARRIED}(\mathit{Mary})$
$\mathit{UNMARRIED}(\mathit{Joe})$

(a) DL-KB $\mathcal{K}$

$\mathit{happy}(X) \leftarrow \mathit{famous}(X), \mathit{WANTS\text{-}TO\text{-}MARRY}(Y, X)$ $[R1]$
$\mathit{RICH}(X) \leftarrow \mathit{famous}(X), not\ \mathit{scientist}(X)$ $[R2]$
$\mathit{famous}(\mathit{Mary})$
$\mathit{famous}(\mathit{Paul})$
$\mathit{famous}(\mathit{Joe})$
$\mathit{scientist}(\mathit{Joe})$

(b) disjunctive Datalog program $\mathcal{P}$

Figure 2: $\mathcal{DL}$+*log*-KB $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ of Example 6

General algorithm

Given a program $\mathcal{P}$, we denote by $\mathcal{C}_\mathcal{P}$ the set of constants occurring in $\mathcal{P}$.

In the following definition, we assume that a rule R in $\mathcal{P}$ has the form $\alpha_R(\vec{x}) \leftarrow \beta_R(\vec{x},\vec{y},\vec{w}), \gamma_R(\vec{x},\vec{y},\vec{z})$, where $\gamma_R(\vec{x},\vec{y},\vec{z})$ is the set of DL-atoms occurring in the body of R (and, of course, $\beta_R(\vec{x},\vec{y},\vec{w})$ is the set of Datalog atoms in the body of R), $\vec{x}$ are the head variables in R, $\vec{y}$ are the existential variables occurring both in DL-atoms and in Datalog atoms in R, and $\vec{z}$ (respectively, $\vec{w}$) are the existential variables of R that only occur in DL-atoms (respectively, Datalog atoms) in R.

Definition 7 *Let $\mathcal{B} = (\mathcal{K},\mathcal{P})$ be a $\mathcal{DL}$+log-KB. The* DL-grounding of $\mathcal{P}$*, denoted by $gr_p(\mathcal{P})$, is the following set of Boolean CQs:*

$$
\begin{aligned}
gr_p(\mathcal{P}) = \{&\gamma_R(\vec{c}_1/\vec{x},\vec{c}_2/\vec{y},\vec{z}) \mid \\
&R \in \mathcal{P} \text{ and } \vec{c}_1,\vec{c}_2 \text{ are tuples of constants in } \mathcal{C}_\mathcal{P}\} \\
&\cup \\
&\{p(\vec{c}/\vec{x}) \mid \\
&\quad p \text{ is a DL-predicate occurring in a rule head} \\
&\quad \text{in } \mathcal{P} \text{ and } \vec{c} \text{ is a tuple of constants in } \mathcal{C}_\mathcal{P}\}
\end{aligned}
$$

Notice that $gr_p(\mathcal{P})$ constitutes a *partial* grounding of the conjunctions of DL-atoms that occur in $\mathcal{P}$ with respect to the constants in $\mathcal{C}_\mathcal{P}$, since the variables that only occur in DL-atoms in the body of rules are not replaced by constants in $gr_p(\mathcal{P})$.

Let G be a set of Boolean CQs. Then, we denote by $CQ(G)$ the Boolean CQ corresponding to the conjunction of all the Boolean CQs in G, i.e., $CQ(G) = \bigwedge_{\gamma\in G}\gamma$. We also denote by $UCQ(G)$ the Boolean UCQ corresponding to the disjunction of all the Boolean CQs in G, namely $UCQ(G) = \bigvee_{\gamma\in G}\gamma$.[3]

Similarly to $gr(\mathcal{P},\mathcal{C}_\mathcal{P})$, we define the *partial grounding of $\mathcal{P}$ on $\mathcal{C}_\mathcal{P}$* (denoted by $pgr(\mathcal{P},\mathcal{C}_\mathcal{P})$) as the program obtained from $\mathcal{P}$ by grounding with the constants in $\mathcal{C}_\mathcal{P}$ all variables *except the existential variables of R that only occur in DL-atoms*.

Finally, given a partition (G_P, G_N) of $gr_p(\mathcal{P})$, we denote by $\mathcal{P}(G_P,G_N)$ the ground Datalog$^{\neg\vee}$ program obtained from $pgr(\mathcal{P},\mathcal{C}_\mathcal{P})$ by:

- deleting all occurrences of the conjunction γ from the body of the rules, for each $\gamma \in G_P$;
- deleting each rule in which γ occurs in the body, for each $\gamma \in G_N$;
- deleting each rule in which γ occurs in the head, for each $\gamma \in G_P$;
- deleting all occurrences of the conjunction γ from the head of the rules, for each $\gamma \in G_N$.

[3]Without loss of generality, we assume that each γ in G uses different existential variable symbols, so that the expression $\bigwedge_{\gamma\in G}\gamma$ can be immediately turned into a Boolean CQ by factoring out all existential quantifications (an analogous simple transformation is needed for turning $UCQ(G)$ into a Boolean UCQ).

Notice that $\mathcal{P}(G_P,G_N)$ is a ground Datalog$^{\neg\vee}$ program over Σ_D, i.e., no DL-predicate occurs in such a program.

We are now ready to present the algorithm NMSAT-$\mathcal{DL}$+*log* for deciding NM-satisfiability of $\mathcal{DL}$+*log*-KBs. The algorithm is shown in Figure 3. The algorithm has a very simple structure, since it decides satisfiability by looking for a guess (G_P,G_N) of the Boolean CQs in $gr_p(\mathcal{P})$ that is consistent with the $\mathcal{DL}$-KB $\mathcal{K}$ and such that the Datalog$^{\neg\vee}$ program $\mathcal{P}(G_P,G_N)$ has a stable model.

Correctness of the algorithm is based on the following property, which relates consistency of a guess (G_P,G_N) of Boolean CQs with the problem of containment of a Boolean CQ in a Boolean UCQ with respect to a $\mathcal{DL}$-TBox.

Lemma 8 *There exists a model $\mathcal{M}$ for $\mathcal{K} = (\mathcal{T},\mathcal{A})$ such that every Boolean CQ in G_P is satisfied in $\mathcal{M}$ and every Boolean CQ in G_N is not satisfied in $\mathcal{M}$ if and only if $\mathcal{T} \not\models CQ(\mathcal{A}\cup G_P) \subseteq UCQ(G_N)$.*

Based on the above lemma, we are able to prove correctness of the algorithm NMSAT-$\mathcal{DL}$+*log*.

Theorem 9 *Let $\mathcal{B}$ be a $\mathcal{DL}$+log-KB. Then, $\mathcal{B}$ is NM-satisfiable iff NMSAT-$\mathcal{DL}$+log$(\mathcal{B})$ returns* true.

Algorithm for *DL-lite+log*

Then, we provide a specialized method for the description logic *DL-lite* (Calvanese *et al.* 2005): more specifically, we study the case of *DL-lite*+*log*-KBs with *positive* Datalog rules (we recall that, by Theorem 3, in this case the FOL semantics and the NM semantics coincide). For such KBs, we are able to define an algorithm that (instead of guessing the truth value of the conjunctions in $gr_p(\mathcal{P})$) generalizes the standard bottom-up computation of the minimal model of a positive Datalog program.

The algorithm is displayed in Figure 4. Basically, at each iteration, the algorithm applies the rules in $pgr(\mathcal{P},\mathcal{C}_\mathcal{P})$: every such rule R is of the form $\alpha \leftarrow \beta,\gamma$, where β is the set of Datalog atoms in the body of R, and γ is the set of DL-atoms in the body of R. If R is "fired", i.e., all the facts in β have already been derived and the Boolean conjunctive query γ is entailed by the initial *DL-lite*-KB $\mathcal{K}$ augmented with the DL-atom derived in the previous iterations, then the atom α (which can be either a DL-atom or a Datalog atom) is derived. The computation is iterated until a fixpoint is reached, i.e., no new facts are derived.

Notice that, in order to check whether a rule is fired, the algorithm has to solve the problem of *answering* a Boolean conjunctive query over a *DL-lite*-KB (i.e., entailment of the conjuctive query γ wrt the *DL-lite*-KB $(\mathcal{T},\mathcal{A}_N)$). Thus, in this algorithm we resort to conjunctive query answering (instead of query containment): notably, very efficient algorithms for answering conjunctive queries have been defined for *DL-lite* (Calvanese *et al.* 2005).

Theorem 10 *Let $\mathcal{B} = (\mathcal{K},\mathcal{P})$ be a DL-lite+log-KB such that $\mathcal{P}$ is a positive Datalog program. $\mathcal{B}$ is FOL-satisfiable (or, equivalently, NM-satisfiable) iff SAT-DL-lite+log$(\mathcal{B})$ returns* true.

Algorithm NMSAT-$\mathcal{DL}$+*log*($\mathcal{B}$)
Input: $\mathcal{DL}$+*log*-KB $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ with $\mathcal{K} = (\mathcal{T}, \mathcal{A})$
Output: true if $\mathcal{B}$ is NM-satisfiable, false otherwise
begin
  if there exists partition (G_P, G_N) of $gr_p(\mathcal{P})$
  such that
    (a) $\mathcal{P}(G_P, G_N)$ has a stable model **and**
    (b) $\mathcal{T} \not\models CQ(\mathcal{A} \cup G_P) \subseteq UCQ(G_N)$
  then return true
  else return false
end

Figure 3: The algorithm NMSAT-$\mathcal{DL}$+*log*

Algorithm SAT-*DL-lite*+*log*($\mathcal{B}$)
Input: *DL-lite*+*log*-KB $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ with $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *DL-lite*-KB,
  $\mathcal{P}$ positive Datalog program with constraints
Output: true if $\mathcal{B}$ is satisfiable, false otherwise
begin
  $\mathcal{A}_N := \mathcal{A}$;
  EDB $:= \emptyset$;
  repeat
    $\mathcal{A}' := \mathcal{A}_N$;
    EDB$'$:= *EDB*;
    for each rule $R \in pgr(\mathcal{P}, \mathcal{C}_\mathcal{P})$ with $R = \alpha \leftarrow \beta, \gamma$ **do**
      if $\beta \in$ *EDB* **and** $(\mathcal{T}, \mathcal{A}_N) \models \gamma$
      then if α is empty (i.e., R is a constraint)
          then return false
          else if α is a DL-atom
              then $\mathcal{A}_N := \mathcal{A}_N \cup \{\alpha\}$
              else *EDB* $:=$ *EDB* $\cup \{\alpha\}$
  until ($\mathcal{A}_N = \mathcal{A}'$) **and** (*EDB* $=$ *EDB*$'$);
  if $(\mathcal{T}, \mathcal{A}_N)$ is a consistent *DL-lite*-KB
  then return true
  else return false
end

Figure 4: The algorithm SAT-*DL-lite*+*log*

Decidability and complexity

First, from the analysis of the algorithm NMSAT-$\mathcal{DL}$+*log* presented above, we are able to prove a very general property that states decidability of reasoning in $\mathcal{DL}$+*log* whenever the Boolean CQ/UCQ containment problem is decidable in $\mathcal{DL}$.

Theorem 11 *For every description logic $\mathcal{DL}$, satisfiability of $\mathcal{DL}$+log-KBs (both under FOL semantics and under NM semantics) is decidable iff Boolean CQ/UCQ containment is decidable in $\mathcal{DL}$.*

From the above theorem and from previous results on query answering and query containment in DLs, we are able to state decidability of reasoning in $\mathcal{DL}$+*log* in the case when $\mathcal{DL}$ corresponds to several known DLs.

In particular, we observe that, for the description logic $\mathcal{DLR}$ (Calvanese *et al.* 1998), it is known that Boolean CQ/UCQ containment is decidable, hence reasoning in $\mathcal{DLR}$+*log*-KBs is decidable.

Theorem 12 *Satisfiability of $\mathcal{DLR}$+log-KBs (both under FOL semantics and under NM semantics) is decidable.*

Since $\mathcal{DLR}$ is a generalization of many expressive DLs (Calvanese *et al.* 1998), this result proves decidability of adding weakly-safe Datalog$^{\neg\vee}$ rules in many DLs.[4]

For the description logic $\mathcal{SHIQ}$ it is known that conjunctive query answering is decidable (see e.g. (Ortiz de la Fuente *et al.* 2005)), but decidability of Boolean CQ/UCQ containment in $\mathcal{SHIQ}$ has not been studied yet, therefore satisfiability in $\mathcal{SHIQ}$+*log* is still an open problem: however, we conjecture that Boolean CQ/UCQ containment in $\mathcal{SHIQ}$ is decidable as well, and hence that reasoning in $\mathcal{SHIQ}$+*log* is decidable.

For *DL-lite*+*log*, besides decidability (which is a corollary of Theorem 12 since $\mathcal{DLR}$ is a generalization of *DL-lite*), we are able to establish the computational complexity of reasoning for different classes of Datalog programs. More precisely, the following theorem refers to *data complexity* of satisfiability, which in the framework of $\mathcal{DL}$+*log* corresponds to the analysis of the computational complexity of the problem when we only consider the size of the ABox $\mathcal{A}$ and of the EDB of $\mathcal{P}$, i.e., the set of facts contained in $\mathcal{P}$. In other words, data complexity considers the TBox $\mathcal{T}$ and the rules not corresponding to facts (i.e., the IDB) in $\mathcal{P}$ as fixed, hence they are not part of the input. Data complexity is a very significant measure when the size of the data, i.e., the ABox and the EDB of $\mathcal{P}$, is much larger than the size of the intensional knowledge, i.e., the TBox and the IDB of $\mathcal{P}$.

The following results are based on the analysis of the previous algorithms and on the fact that conjunctive query answering in *DL-lite* is in PTIME in data complexity (actually it is in LOGSPACE) (Calvanese *et al.* 2005).

Theorem 13 *Let $\mathcal{B} = (\mathcal{K}, \mathcal{P})$ be a DL-lite+log-KB. Then:*

- *if $\mathcal{P}$ is a positive Datalog program, then deciding FOL-satisfiability (or, equivalently, NM-satisfiability) of $\mathcal{B}$ is PTIME-complete with respect to data complexity;*
- *if $\mathcal{P}$ is a positive disjunctive Datalog program, then deciding FOL-satisfiability (or, equivalently, NM-satisfiability) of $\mathcal{B}$ is NP-complete with respect to data complexity;*
- *if $\mathcal{P}$ is an arbitrary Datalog$^{\neg\vee}$ program, then deciding NM-satisfiability of $\mathcal{B}$ is Σ_2^p-complete with respect to data complexity.*

Therefore, in DL-lite, under both semantics, the data complexity does not increase with respect to the data complexity of the Datalog program alone. In other words, connecting a *DL-lite*-KB to a Datalog program does not increase complexity of reasoning in the size of the data. We also point out that *DL-lite* with arbitrary, non-weakly-safe recursive Datalog rules is undecidable (which follows from the results in (Levy & Rousset 1998; Calvanese & Rosati 2003)).

Related work

As mentioned above, several recent studies propose various forms of integration between DLs and rules. The first formal proposals for the integration of Description Logics and rules are $\mathcal{AL}$-log (Donini *et al.* 1998) and CARIN (Levy & Rousset 1996a; 1996b; 1998).

$\mathcal{AL}$-log is a framework which integrates KBs expressed in the description logic $\mathcal{ALC}$ and positive Datalog programs: the interaction between the DL-KB and the rules is controlled by a syntactic condition that corresponds to the DL-safeness above mentioned, which states that every variable of a rule R must appear in at least one of the Datalog atoms occurring in the body of R.

Research in *non-safe* interaction of DLs and rules was started by the work on CARIN (Levy & Rousset 1996a; 1996b; 1998), which established very important undecidability results concerning non-safe interaction between DL-KBs and rules. Roughly speaking, such results clearly indicate that, in case of unrestricted interaction between DL-KBs and rules, decidability of reasoning holds only if at least one of the two components has very limited expressive power: e.g., in order to retain decidability of reasoning, allowing recursion in rules imposes very severe restrictions on the expressiveness of the DL-KB.

Then, $\mathcal{DL}$-*log* was proposed in (Rosati 1999) 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, while keeping the above DL-safeness condition on variables. This framework has been further extended in (Rosati 2005a) to the integration of arbitrary, decidable, first-order theories and disjunctive Datalog rules based on an analogous notion of safeness.

The framework of $\mathcal{AL}$-log has been extended in a different way in (Motik, Sattler, & Studer 2004; 2005). There, the problem of extending OWL-DL with positive Datalog programs is analyzed. The interaction between OWL-DL and rules is again restricted through the DL-safeness condition above described. With respect to $\mathcal{DL}$-*log*, in (Motik, Sattler, & Studer 2005) a more expressive structural language and

[4] The first DL for which it was proved that Boolean CQ/UCQ containment is decidable is $\mathcal{ALCNR}$, studied in (Levy & Rousset 1998), which actually corresponds to a restricted version of $\mathcal{DLR}$.

a less expressive rule language are adopted: moreover, the information flow is bidirectional, i.e., structural predicates may appear in the head of rules.

The work presented in (Grosof *et al.* 2003) 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.

Another approach for extending DLs with Datalog$^\neg$ rules is presented in (Eiter *et al.* 2004a; 2004b). Differently from $\mathcal{DL}$+*log* 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. In particular, a condition equivalent to the DL-safeness on variables is imposed at the semantic level (rather than by the syntax), since the meaning of every rule correspond to the grounding of such rule over the constants occurring in the program. Thus, from the semantic viewpoint, this form of interaction-via-entailment between the DL-KB and the rules is more restricted than the interaction provided by $\mathcal{DL}$+*log*. On the other hand, such an increased separation in principle allows for more modular reasoning techniques, which are able to completely separate reasoning about the DL-KB and reasoning about rules.

Finally, another recent proposals in this field is SWRL (Horrocks & Patel-Schneider 2004), a non-safe approach to the integration of rules and DL-KBs in which rules are interpreted under the classical FOL semantics. The addition of this kind of rules to DLs leads to undecidability of reasoning.

Discussion

The present work aims at extending the integration of DLs and Datalog based on the DL-safeness condition recalled in the previous section, which is actually adopted (although through different formal assumptions) by many of the proposals previously mentioned (Donini *et al.* 1998; Eiter *et al.* 2004a; Motik, Sattler, & Studer 2005; Rosati 2005a).

From the viewpoint of the expressive power, $\mathcal{DL}$+*log* provides a significant improvement with respect to the approaches based on DL-safe rules. Indeed, simple non-recursive queries over $\mathcal{DL}$-KBs, like conjunctive queries and unions of conjunctive queries (which are computable and for which there are known algorithms in many DLs), cannot be fully expressed in terms of DL-safe rules, because of the presence of existential variables in a conjunctive query: imposing DL-safeness over such existential variables drastically changes the meaning of the query.

Example 14 Let $\mathcal{K}$ be a $\mathcal{DL}$-KB over the concepts C, D and the roles R, S, T. Let q be the following Boolean union of conjunctive queries:

$$q = \exists x, y, z.\, C(x), R(x, y), C(y), R(y, z), C(z), R(z, x) \\ \vee\, D(x), S(x, y), T(y, x)$$

The query q can be correctly formalized by the following program $\mathcal{P}$ consisting of the following weakly-safe rules:

$$\leftarrow C(X), R(X, Y), C(Y), R(Y, Z), C(Z), R(Z, X)$$
$$\leftarrow D(X), S(X, Y), T(Y, X)$$

Indeed, it is immediate to see that $\mathcal{K} \models q$ iff $(\mathcal{K}, \mathcal{P})$ is unsatisfiable. Notice that the above rules are weakly-safe (and hence expressible in $\mathcal{DL}$+*log*) but not DL-safe: to satisfy DL-safeness, one should force, in each rule, all the existential variables to occur in auxiliary atoms which restrict such variables to range only on the constants explicitly mentioned in the ABox, thus changing the meaning of the original query q. ∎

On the other hand, it can be shown that, in the case of (non-Boolean) CQs and UCQs with head variables, the safeness condition on the head variables does not actually affect the meaning of such queries, since it is commonly assumed that the answers returned by queries must be tuples of constants occurring in the DL-KB. Therefore, CQs and UCQs over $\mathcal{DL}$-KBs can be correctly represented as weakly-safe rules in $\mathcal{DL}$+*log*.

From the reasoning viewpoint, we have proved that in $\mathcal{DL}$+*log* we can actually define reasoning techniques in a way in principle similar to what has been done in the case of DL-safe rules (Motik, Sattler, & Studer 2005; Rosati 2005a). However, in $\mathcal{DL}$+*log* it is harder to arrive at a separation between rules and DL-KB in the reasoning process, in the following sense:

- In the presence of DL-safe rules, the separation can be done by using essentially the traditional reasoning services offered by DL-KBs, in particular, KB satisfiability.
- For $\mathcal{DL}$+*log*, it turns out that we can separate rules from the $\mathcal{DL}$-KB only if the DL-component offers a query containment reasoning service.[5]

In other words, while DL-safeness allows for reducing reasoning in DLs with rules to reasoning in DLs through satisfiability, the weaker notion of safeness of $\mathcal{DL}$+*log* allows for reducing reasoning in DLs with rules to reasoning in DLs through conjunctive query containment.

Conclusions

In this paper we have presented $\mathcal{DL}$+*log*, a general framework for the integration of Description Logics and disjunctive Datalog.

The main features of $\mathcal{DL}$+*log* can be summarized as follows:

- $\mathcal{DL}$+*log* provides a clear formal account of the closed-world semantics of nonmonotonic rules and the open-world semantics of DLs;

[5] We recall that, in general, query containment cannot be reduced to the standard reasoning services offered by a DL.

- through the notion of weak safeness, $\mathcal{DL}$+*log* overcomes the expressive limitations of the approaches to the integration of DLs and rules based on the DL-safeness condition;
- under general conditions, the weakly-safe integration of DLs and Disjunctive Datalog provided by $\mathcal{DL}$+*log* preserves decidability (and complexity) of reasoning;
- reasoning in $\mathcal{DL}$+*log* can be done by composing, in a simple way, reasoning about the DL-KB and reasoning about rules.

As for further work, we aim at studying decidability of $\mathcal{DL}$+*log* for DLs more expressive than $\mathcal{DLR}$ (e.g., OWL-DL), and, more generally, establishing more general computational properties for $\mathcal{DL}$+*log*. To this purpose, a necessary, preliminary step is to find new results on query answering and query containment for unions of conjunctive queries in Description Logics.

Then, another open issue is whether it is possible to identify tighter forms of decidable interaction between DL-KBs and rules, which are able to overcome the limitations of $\mathcal{DL}$+*log*. In this respect, we believe that one of the most important expressive limitations of $\mathcal{DL}$+*log* is the rigid separation between DL-predicates and Datalog predicates: since DL-predicates have an open interpretation while Datalog predicates have a closed interpretation, in $\mathcal{DL}$+*log* it is not possible to express more complex pieces of information in which the same predicate is interpreted in different ways (i.e., both under an open-world assumption and under a closed-world assumption) in different parts of the same knowledge base.

Finally, it would be worth studying optimization of algorithms for *DL-lite+log*, which appears as a very attractive combination of DLs and Datalog, due to the good computational properties shown in this paper.

Acknowledgments

This research has been partially supported by the projects TONES (FP6-7603) and INTEROP Network of Excellence (IST-508011) funded by the EU, by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by MIUR FIRB 2005 project "Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet" (TOCAI.IT). The author wishes to thank Enrico Franconi, Ian Horrocks, Boris Motik, Sergio Tessaris, and the anonymous reviewers for their precious comments.

References

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Calvanese, D., and Rosati, R. 2003. Answering recursive queries under keys and foreign keys is undecidable. In *Proc. of the 10th Int. Workshop on Knowledge Representation meets Databases (KRDB 2003)*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-79/`.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Nardi, D.; and Rosati, R. 1998. Description logic framework for information integration. In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, 2–13.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. DL-Lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, 602–607.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1998. On the decidability of query containment under constraints. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, 149–158.

Donini, F. M.; Lenzerini, M.; Nardi, D.; and Schaerf, A. 1998. $\mathcal{AL}$-log: Integrating Datalog and description logics. *J. of Intelligent Information Systems* 10(3):227–252.

Eiter, T.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2004a. Combining answer set programming with description logics for the semantic web. In *Proc. of the 9th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2004)*, 141–151.

Eiter, T.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2004b. Well-founded semantics for description logic programs in the semantic web. In *Proceedings of the Third International Workshop on Rules and Rule Markup Languages for the Semantic Web (RuleML 2004)*, 81–97.

Eiter, T.; Gottlob, G.; and Mannilla, H. 1997. Disjunctive Datalog. *ACM Trans. on Database Systems* 22(3):364–418.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385.

Grosof, B. N.; Horrocks, I.; Volz, R.; and Decker, S. 2003. Description logic programs: Combining logic programs with description logic. In *Proc. of the 12th Int. World Wide Web Conf. (WWW 2003)*, 48–57.

Horrocks, I., and Patel-Schneider, P. F. 2004. A proposal for an OWL rules language. In *Proceedings of the 13th international conference on World Wide Web (WWW 2004)*, 723–731.

Levy, A. Y., and Rousset, M.-C. 1996a. CARIN: A representation language combining Horn rules and description logics. In *Proc. of the 12th Eur. Conf. on Artificial Intelligence (ECAI'96)*, 323–327.

Levy, A. Y., and Rousset, M.-C. 1996b. The limits on combining recursive Horn rules with description logics. In *Proc. of the 13th Nat. Conf. on Artificial Intelligence (AAAI'96)*, 577–584.

Levy, A. Y., and Rousset, M.-C. 1998. Combining Horn rules and description logics in CARIN. *Artificial Intelligence* 104(1–2):165–209.

Motik, B.; Sattler, U.; and Studer, R. 2004. Query answering for OWL-DL with rules. In *Proceedings of the 2004 International Semantic Web Conference (ISWC 2004)*, 549–563.

Motik, B.; Sattler, U.; and Studer, R. 2005. Query answering for OWL-DL with rules. *Web Semantics* 3(1):41–60.

Ortiz de la Fuente, M. M.; Calvanese, D.; Eiter, T.; and Franconi, E. 2005. Data complexity of answering conjunctive queries over $\mathcal{SHIQ}$ knowledge bases. Technical report, Faculty of Computer Science, Free University of Bozen-Bolzano. Also available as CORR technical report at `http://arxiv.org/abs/cs.LO/0507059/`.

Rosati, R. 1999. Towards expressive KR systems integrating Datalog and description logics: Preliminary report. In *Proc. of the 1999 Description Logic Workshop (DL'99)*, 160–164. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-22/`.

Rosati, R. 2005a. On the decidability and complexity of integrating ontologies and rules. *Web Semantics* 3(1):61–73.

Rosati, R. 2005b. Semantic and computational advantages of the safe integration of ontologies and rules. In *Proceedings of the 2005 International Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR 2005)*, volume 3703 of *Lecture Notes in Computer Science*, 50–64. Springer.