

On Combining Description Logic Ontologies and Nonrecursive Datalog Rules

Riccardo Rosati

Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma, Italy
rosati@dis.uniroma1.it

Abstract. Reasoning in systems integrating Description Logics (DL) ontologies and Datalog rules is a very hard task, and previous studies have shown undecidability of reasoning in systems integrating (even very simple) DL ontologies with recursive Datalog. However, the results obtained so far constitute a very partial picture of the computational properties of systems combining DL ontologies and Datalog rules. The aim of this paper is to contribute to complete this picture, extending the computational analysis of reasoning in systems integrating ontologies and Datalog rules. More precisely, we first provide a set of decidability and complexity results for reasoning in systems combining ontologies specified in DLs and rules specified in *nonrecursive* Datalog (and its extensions with inequality and negation): such results identify, from the viewpoint of the expressive abilities of the two formalisms, minimal combinations of Description Logics and Datalog in which reasoning is undecidable. Then, we present new results on the decidability and complexity of the so-called *restricted* (or *safe*) integration of DL ontologies and Datalog rules. Our results show that: (1) the unrestricted interaction between DLs and Datalog is computationally very hard even in the absence of recursion in rules; (2) surprisingly, the various "safeness" restrictions, which have been defined to regain decidability of reasoning in the interaction between DLs and recursive Datalog, appear as necessary restrictions even when rules are not recursive.

1 Introduction

Background The problem of adding rules to ontologies is currently a hot research topic, due to the interest of Semantic Web applications towards the integration of rule-based systems with ontologies. Most of the approaches in this field concern the study of Description Logic (DL) knowledge bases [3] augmented with rules expressed in Datalog and its nonmonotonic extensions [9].

DLs 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 [26]. DLs are a family of knowledge representation formalisms based on first-order logic (FOL). In fact, almost all DLs coincide with decidable fragments of function-free first-order logic with equality, and the language of a DL can be seen as a restricted FOL language over unary and binary predicates and with a controlled form of quantification (actually, DLs are equipped with a special, variable-free syntax). Notably, DLs have been designed to optimize the trade-off between expressive abilities and complexity of

reasoning, hence the computational properties of DLs have been extensively studied [3].

From the knowledge representation viewpoint, Datalog is somehow “complementary” to DLs. Indeed, with respect to DLs, Datalog allows for using predicates of arbitrary arity, the explicit use of variables, and the ability of expressing more powerful queries. Moreover, its nonmonotonic features (in particular, the negation-as-failure operator *not*) allow for expressing default rules and forms of closed-world reasoning.

Problem studied Unfortunately, reasoning in systems integrating DLs and Datalog is a very hard task, and well-known previous results have shown undecidability of reasoning in systems fully integrating (even very simple) DL ontologies with Datalog rules. In fact, in general this combination does not preserve decidability, i.e., starting from a DL knowledge base in which reasoning is decidable and a set of rules in which reasoning is decidable, reasoning in the knowledge base obtained by integrating these two components may not be a decidable problem.

To avoid undecidability of reasoning, practically all decidable approaches to integrating ontologies and rules impose (either at the syntactic or at the semantic level) specific conditions which restrict the interaction between the rules and the ontology. Such restrictions were mainly introduced to keep reasoning decidable in the presence of recursion in Datalog rules.

However, the results obtained so far [20, 11, 18, 23, 27, 28, 10] actually constitute a very partial picture of the computational properties of systems combining DL ontologies and Datalog rules. In particular, the computational properties of systems combining DL ontologies and the class of *nonrecursive* Datalog rules are mostly unknown. The only known studies related to this topic are the work on CARIN [20], which has shown decidability of nonrecursive positive Datalog with the DL $\mathcal{ALCN}\mathcal{R}$, and the studies on conjunctive query answering in DLs (see e.g. [7, 24, 25, 14, 15]), which are indirectly related to integrating Datalog and DLs (since conjunctive queries can be seen as nonrecursive Datalog programs consisting of a single rule).

Contribution The aim of this paper is to contribute to fill this gap, extending the computational analysis of reasoning in systems integrating ontologies and Datalog rules. More precisely, our contributions can be summarized as follows:

- We first provide a set of decidability and complexity results for reasoning in systems combining ontologies specified in (different classes of) DLs and rules specified in (different classes of) *nonrecursive* Datalog (and its extensions with inequality or negation). Such results identify, from the viewpoint of the expressive abilities of the two formalisms, minimal combinations of Description Logics and (non-monotonic) Datalog in which reasoning is undecidable. A summary of the results obtained is reported in Figure 2 (Section 4).
- Then, we present new results on the decidability and complexity of the restricted integration of DL ontologies and Datalog rules. In particular, we consider the so-called “weakly DL-safe” interaction between rules and DL ontologies [28], which is currently one of the most expressive decidable combinations of DLs and rules:

we extend the framework of [28] to deal with both negation of DL predicates and the presence of inequality, and provide new decidability and complexity results for such a class of weakly DL-safe Datalog rules.

Besides constituting one of the first refined computational analyses taking into account the expressive power of both the DL language and the rule language (the only similar study which we are aware of is [20]), the above results imply the following consequences:

- the unrestricted interaction of DLs and Datalog is computationally very hard even in the absence of recursion in rules. This contrasts with the general opinion (suggested by the results in [20]) that the presence of recursion in rules is necessary in order to rise the undecidability issue in their combination with DL ontologies;
- surprisingly, the “safeness” restrictions, which have been defined to regain decidability in the interaction between DLs and recursive Datalog, appear as necessary restrictions even when rules are not recursive.

Structure of the paper In Section 2, we briefly recall the basics of Description Logics and Datalog. In Section 3, we formally define syntax and semantics of systems integrating DLs and Datalog. In Section 4, we consider the full integration of DLs and rules, and present a set of undecidability and hardness results for reasoning in systems fully combining DLs and Datalog rules. In Section 5, we focus on weakly DL-safe systems, which are based on a restricted form of interaction between DLs and rules, extend them to the presence of inequality atoms, and present a computational analysis of reasoning in such systems. Finally, we conclude in Section 6. Due to space limits, in the present version of the paper we provide proof sketches of the theorems.

2 Description Logics and Datalog

In this section we briefly introduce Description Logics and Datalog.

Description Logics We now briefly recall the basics of Description Logics (DLs) and introduce the following DLs: (i) three prominent *tractable* DLs, i.e., $DL-Lite_{RDFS}$, $DL-Lite_R$ and \mathcal{EL} ; (ii) the “classical” and *moderately expressive* DL \mathcal{ALC} ; (iii) two *very expressive* DLs, i.e., \mathcal{SHIQ} and \mathcal{DLR} . We refer to [3] for a more detailed introduction to DLs.

We start from an alphabet of concept names, an alphabet of role names and an alphabet of constant names. Concepts correspond to unary predicates in FOL, roles correspond to binary predicates, and constants corresponds to FOL constants.

Starting from concept and role names, *concept expressions* and *role expressions* can be constructed, based on a formal syntax. Different DLs are based on different languages concept and role expressions. Details on the concept and role languages for the DLs considered in this paper are reported below.

A *concept inclusion* is an expression of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concept expressions. Similarly, a *role inclusion* is an expression of the form $R_1 \sqsubseteq R_2$, where R_1 and R_2 are role expressions.

DL	concept expressions	role expressions	TBox axioms
$DL-Lite_{RDFS}$	$C_L ::= A \mid \exists R$ $C_R ::= A$	$R ::= P \mid P^-$	$C_L \sqsubseteq C_R$ $R_1 \sqsubseteq R_2$
$DL-Lite_R$	$C_L ::= A \mid \exists R$ $C_R ::= A \mid \neg C_R \mid \exists R$	$R ::= P \mid P^-$	$C_L \sqsubseteq C_R$ $R_1 \sqsubseteq R_2$
\mathcal{EL}	$C ::= A \mid C_1 \sqcap C_2 \mid \exists P.C$	$R ::= P$	$C_1 \sqsubseteq C_2$
\mathcal{ALC}	$C ::= A \mid C_1 \sqcap C_2 \mid \neg C \mid \exists P.C$	$R ::= P$	$C_1 \sqsubseteq C_2$
\mathcal{SHIQ}	$C ::= A \mid \neg C \mid C_1 \sqcap C_2 \mid (\geq n RC)$	$R ::= P \mid P^-$	$C_1 \sqsubseteq C_2$ $R_1 \sqsubseteq R_2$ $\text{Trans}(R)$

Fig. 1. Abstract syntax of the DLs studied in the paper.

An *instance assertion* is an expression of the form $A(a)$ or $P(a, b)$, where A is a concept name, P is a role name, and a, b are constant names. We do not consider complex concept and role expressions in instance assertions, since in this paper we are interested in data complexity of reasoning (see Section 4).

A *DL knowledge base (KB)* is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$, where \mathcal{T} , called the *TBox*, is a set of concept and role inclusions, and \mathcal{A} , called the *ABox*, is a set of instance assertions.

The DLs mainly considered in this paper are the following:

- $DL-Lite_{RDFS}$, which corresponds to the “DL fragment” of RDFS [1], the schema language for RDF (see also [16]);
- $DL-Lite_R$ [5], a tractable DL which is tailored for efficient reasoning and query answering in the presence of very large ABoxes;
- \mathcal{EL} [2], a prominent tractable DL;
- \mathcal{ALC} , a very well-known DL which corresponds to multimodal logic K_n [3];
- \mathcal{SHIQ} , a very expressive DL which constitutes the basis of the OWL family of DLs adopted as standard languages for ontology specification in the Semantic Web [26].

The syntax of the above DLs is summarized in Figure 1, in which the symbol A denotes a concept name and the symbol P denotes a role name (in addition to concept and role inclusions, \mathcal{SHIQ} also allows for TBox axioms of the form $\text{Trans}(R)$, which state transitivity of the role R).

We will also mention the DL \mathcal{DLR} [7], which informally extends \mathcal{SHIQ} (without transitive roles) through the use of n -ary relations, and for which decidability results on query answering are known (we refer to [7] for details on the syntax of \mathcal{DLR} , which is quite different from the other DLs due to the usage of relations of arbitrary arity).

The above mentioned DLs verify the following ordering with respect to their relative expressive power (see [3] for details): $DL-Lite_{RDFS} \subset DL-Lite_R \subset \mathcal{SHIQ} \subset \mathcal{DLR}$; and $\mathcal{EL} \subset \mathcal{ALC} \subset \mathcal{SHIQ}$.

We give the semantics of DLs through the well-known translation ρ_{fol} of DL knowledge bases into FOL theories with counting quantifiers (see [3]).

$$\begin{aligned}
\rho_{fol}(\langle \mathcal{T}, \mathcal{A} \rangle) &= \rho_{fol}(\mathcal{T}) \cup \rho_{fol}(\mathcal{A}) \\
\rho_{fol}(C_1 \sqsubseteq C_2) &= \forall x. \rho_{fol}(C_1, x) \rightarrow \rho_{fol}(C_2, x) \\
\rho_{fol}(R_1 \sqsubseteq R_2) &= \forall x, y. \rho_{fol}(R_1, x, y) \rightarrow \rho_{fol}(R_2, x, y) \\
\rho_{fol}(\text{Trans}(R)) &= \forall x, y, z. \rho_{fol}(R, x, y) \wedge \rho_{fol}(R, y, z) \rightarrow \rho_{fol}(R, x, z) \\
\rho_{fol}(A, x) &= A(x) \\
\rho_{fol}(\neg C, x) &= \neg \rho_{fol}(C, x) \\
\rho_{fol}(C_1 \sqcap C_2, x) &= \rho_{fol}(C_1, x) \wedge \rho_{fol}(C_2, x) \\
\rho_{fol}(\exists R, x) &= \exists y. \rho_{fol}(R, x, y) \\
\rho_{fol}(\exists R.C, x) &= \exists y. \rho_{fol}(R, x, y) \wedge \rho_{fol}(C, y) \\
\rho_{fol}(\geq n R C, x) &= \exists \geq n y. \rho_{fol}(R, x, y) \wedge \rho_{fol}(C, y) \\
\rho_{fol}(P, x, y) &= P(x, y) \\
\rho_{fol}(P^-, x, y) &= P(y, x)
\end{aligned}$$

An interpretation of \mathcal{K} is a classical FOL interpretation for $\rho_{fol}(\mathcal{K})$, where constants and predicates are interpreted over a non-empty interpretation domain which may be finite or countably infinite. Actually, in this paper we adopt the standard names assumption, i.e.: (i) we assume a countably infinite set of constant symbols Γ ; (ii) the interpretation domain Δ is countably infinite and is the same for every interpretation; (iii) the interpretation of constants in Γ is the same in every interpretation and is given by a one-to-one correspondence between Γ and Δ . Such an assumption is necessary for the nonmonotonic semantics defined in Section 3: however, we point out that all the results presented in this paper under the first-order semantics (i.e., the results for FOL-satisfiability) also hold in the absence of the standard names assumption.

A *model* of a DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a FOL model of $\rho_{fol}(\mathcal{K})$. We say that \mathcal{K} is *satisfiable* if \mathcal{K} has a model.

Disjunctive Datalog In this section we briefly recall disjunctive Datalog [9], denoted by $\text{Datalog}^{-\vee}$, which is the well-known nonmonotonic extension of Datalog with negation as failure and disjunction.

We start from a predicate alphabet, a constant alphabet, and a variable alphabet. 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*). A $\text{Datalog}^{-\vee}$ rule R is an expression of the form

$$\alpha_1 \vee \dots \vee \alpha_n \leftarrow \beta_1, \dots, \beta_m, \text{not } \gamma_1, \dots, \text{not } \gamma_k, t_1 \neq t'_1, \dots, t_h \neq t'_h \quad (1)$$

where each $\alpha_i, \beta_i, \gamma_i$ is an atom, each t_i, t'_i is either a variable or a constant, and every variable occurring in R must appear in at least one of the atoms β_1, \dots, β_m . This last condition is known as the *Datalog safeness* condition for variables. The variables occurring in the atoms $\alpha_1, \dots, \alpha_n$ are called the *head variables* of R . If $n = 0$, we call R a *constraint*.

A $\text{Datalog}^{-\vee}$ program is a set of $\text{Datalog}^{-\vee}$ rules. If, for all $R \in \mathcal{P}$, $k = 0$ and $h = 0$, \mathcal{P} is called a *positive disjunctive Datalog* program. If, for all $R \in \mathcal{P}$, $n \leq 1$, $k = 0$ and $h = 0$, \mathcal{P} is called a *positive Datalog* program. If there are no occurrences

of variable symbols in a rule R , then R is called a *ground* rule. A *ground* program is a program containing only ground rules.

The *dependency graph* of a program P is a graph whose nodes are the predicates of P and in which there is an edge from p_1 to p_2 if there is a rule r in P such that p_2 occurs in the body of r and p_1 occurs in the head of r . A program P is *recursive* if its dependency graph contains a cycle. Otherwise, P is called *nonrecursive*.

The semantics of disjunctive Datalog is given in terms of *stable models* of a program \mathcal{P} . Due to space limitations, we refer to [9] for details on such semantics: however, in the following we will provide a detailed definition of such semantics in the more general framework of r-hybrid KBs integrating DLs and disjunctive Datalog.

3 R-hybrid KBs

In this section we present the framework of r-hybrid KBs which integrate DLs with disjunctive Datalog. More precisely, we slightly extend the framework of r-hybrid knowledge bases presented in [27] to the presence of both inequality atoms and negation of DL predicates in rules.

Syntax From the syntactic viewpoint, integrating a DL with (disjunctive) Datalog simply means the possibility of writing a *hybrid* knowledge base \mathcal{H} containing a DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a disjunctive Datalog program \mathcal{P} (i.e., $\mathcal{H} = (\mathcal{K}, \mathcal{P})$) where \mathcal{K} and \mathcal{P} share both the alphabet of predicates and the alphabet of constants. However, for technical reasons related to the subsequent definition of the nonmonotonic semantics, we distinguish the predicates occurring only in \mathcal{P} , which we call *Datalog predicates*, from the ones occurring both in \mathcal{K} and in \mathcal{P} , or even only in \mathcal{K} , which we call *DL predicates*. In the following, we denote by $\Sigma_C \cup \Sigma_R$ the set of DL predicates, and denote by Σ_D the set of Datalog predicates. Formally, a rule R in \mathcal{P} is a rule of the form (1) over both DL-predicates and Datalog predicates. An atom whose predicate is a DL predicate is called a *DL atom*, while an atom whose predicate is a Datalog predicate is called a *Datalog atom*.

First-order semantics According to a semantic approach based on classical logic, the hybrid knowledge base can be considered as a first-order theory, by interpreting Datalog rules as first-order implications. More specifically, let R be the Datalog^{- \forall} rule of the form (1). Then, we denote by $FO(R)$ the first-order sentence

$$\forall x_1, \dots, x_p. \beta_1 \wedge \dots \wedge \beta_m \wedge \neg \gamma_1 \wedge \dots \wedge \neg \gamma_k \wedge t_1 \neq t'_1 \wedge \dots \wedge t_h \neq t'_h \rightarrow \alpha_1 \vee \dots \vee \alpha_n$$

where x_1, \dots, x_p are all the variable symbols appearing in R . Given a Datalog^{- \forall} program \mathcal{P} , we denote by $FO(\mathcal{P})$ the set of first-order sentences $\{FO(R) \mid R \in \mathcal{P}\}$.

Finally, the semantics of a knowledge base $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ composed of a DL-KB \mathcal{K} and a Datalog program \mathcal{P} is given by the first-order theory $FO(\mathcal{H})$ corresponding to the union of $FO(\mathcal{P})$ and the first-order translation $FO(\mathcal{K})$ of \mathcal{K} : in particular, we say that \mathcal{H} is *FOL-satisfiable* if $FO(\mathcal{H})$ has a model (which is called *FOL-model* of \mathcal{H}), and we say that a ground atom g is FOL-entailed by \mathcal{H} , denoted by $\mathcal{H} \models_{FOL} g$ iff, for each FOL-model \mathcal{I} of \mathcal{H} , \mathcal{I} satisfies g .

Nonmonotonic semantics We now recall the nonmonotonic semantics for r-hybrid KBs presented in [27], which is a “conservative extension” of both the open-world semantics (classical FOL models) of DLs and the closed-world semantics (stable models) of disjunctive Datalog.

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

The *ground instantiation* of \mathcal{P} , denoted by $gr(\mathcal{P})$, 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 Γ .

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 $t^\mathcal{I} \in r^\mathcal{I}$;
- delete each atom $r(t)$ in the head of R such that $r \in \Sigma'$ and $t^\mathcal{I} \notin r^\mathcal{I}$;
- delete R if: either (i) there exists an atom $r(t)$ in the body of R such that $r \in \Sigma'$ and $t^\mathcal{I} \notin r^\mathcal{I}$; or (ii) there exists a negated atom $not\ r(t)$ in the body of R such that $r \in \Sigma'$ and $t^\mathcal{I} \in r^\mathcal{I}$;
- delete each atom $r(t)$ in the body of R such that $r \in \Sigma'$ and $t^\mathcal{I} \in r^\mathcal{I}$;
- delete each negated atom $not\ r(t)$ in the body of R such that $r \in \Sigma'$ and $t^\mathcal{I} \notin r^\mathcal{I}$.

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 $\mathcal{I}_1, \mathcal{I}_2$ of the set of predicates Σ , we write $\mathcal{I}_1 \subset_\Sigma \mathcal{I}_2$ if (i) for each $p \in \Sigma$ and for each tuple t of constants from Γ , if $t^{\mathcal{I}_1} \in p^{\mathcal{I}_1}$ then $t^{\mathcal{I}_2} \in p^{\mathcal{I}_2}$, and (ii) there exist $p \in \Sigma$ and tuple t of constants from Γ such that $t^{\mathcal{I}_1} \notin p^{\mathcal{I}_1}$ and $t^{\mathcal{I}_2} \in p^{\mathcal{I}_2}$.

Given a positive disjunctive ground Datalog^{-v} 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: (i) \mathcal{I} satisfies the first-order translation $FO(\mathcal{P})$ of \mathcal{P} ; (ii) 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^{-v} program \mathcal{P} and an interpretation \mathcal{I} for \mathcal{P} , the *GL-reduct* [12] of \mathcal{P} with respect to \mathcal{I} , denoted by $GL(\mathcal{P}, \mathcal{I})$, is the positive disjunctive ground program obtained from \mathcal{P} as follows. For each rule $R \in \mathcal{P}$:

1. delete R if either there exists a negated atom $not\ r(t)$ in the body of R such that $t^\mathcal{I} \in r^\mathcal{I}$, or there exists an inequality $c \neq c$ in the body of R ;
2. delete each negated atom $not\ r(t)$ in the body of R such that $t^\mathcal{I} \notin r^\mathcal{I}$ and delete each inequality $c \neq d$ where c and d are distinct constants.

Given a ground Datalog^{-v} 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 1. An interpretation \mathcal{I} of $\Sigma_C \cup \Sigma_R \cup \Sigma_D$ is a NM-model for $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ if the following conditions hold: (i) $\mathcal{I}_{\Sigma_C \cup \Sigma_R}$ satisfies \mathcal{K} ; (ii) \mathcal{I}_{Σ_D} is a stable model for $\Pi(\text{gr}(\mathcal{P}), \mathcal{I}_{\Sigma_C \cup \Sigma_R})$. \mathcal{H} is called NM-satisfiable if \mathcal{H} has at least one NM-model.

We say that a ground atom g is NM-entailed by \mathcal{H} , denoted by $\mathcal{H} \models_{NM} g$ iff, for each NM-model \mathcal{I} of \mathcal{H} , \mathcal{I} satisfies g .

According to the above semantics, DL predicates are interpreted under the open-world assumption, while Datalog predicates are interpreted under the closed-world assumption of disjunctive Datalog and Answer Set Programming. As a consequence, negation of DL predicates in rule bodies is interpreted as classical (monotonic) negation, while negation of Datalog predicates is interpreted as nonmonotonic negation (negation as failure under stable model semantics).

Reasoning: general properties Notice that, under the above NM semantics (as well as under the FOL semantics), entailment can be reduced to unsatisfiability, since it is possible to express constraints (i.e., rules with empty head) in the Datalog program. More precisely, the following property holds.

Proposition 1. Let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r -hybrid KB and let g be a ground atom. Then, $\mathcal{H} \models_{NM} g$ (respectively, $\mathcal{H} \models_{FOL} g$) iff the r -hybrid KB $(\mathcal{K}, \mathcal{P} \cup \{\leftarrow g\})$ is NM-unsatisfiable (respectively, FOL-unsatisfiable).

Then, we show that, when there are no negated Datalog atoms in the bodies of rules, the above two semantics are equivalent with respect to the satisfiability problem. The following property extends an analogous one shown in [28].

Proposition 2. Let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r -hybrid KB, where \mathcal{P} is such that there are no occurrences of negated Datalog atoms in \mathcal{P} . Then, \mathcal{H} is FOL-satisfiable iff \mathcal{H} is NM-satisfiable.

4 Results for nonrecursive rules

In this section we present a set of new results on the decidability and complexity of reasoning in r -hybrid KBs, under both FOL-semantics and NM-semantics.

We have conducted our computational analysis on the following subclasses of *non-recursive* and *nondisjunctive* Datalog programs:

- *NR-Datalog* = nonrecursive positive Datalog, i.e., nonrecursive rules of the form (1) where $n \leq 1, k = 0, h = 0$;
- *NR-Datalog[≠]* = nonrecursive positive Datalog with inequality, i.e., nonrecursive rules of the form (1) where $n \leq 1, k = 0$;
- *NR-Datalog_s[≠]* = single-rule nonrecursive positive Datalog with inequality (i.e., *NR-Datalog[≠]* programs consisting of a single rule);
- *NR-Datalog[¬]* = nonrecursive Datalog with negation, i.e., nonrecursive rules of the form (1) where $n \leq 1, h = 0$;

- $NR\text{-Datalog}_s^\neg$ = single-rule nonrecursive Datalog with negation (i.e., $NR\text{-Datalog}_s^\neg$ programs consisting of a single rule);
- $NR\text{-Datalog}_s^{\neg A}$ = nonrecursive Datalog with “atomic” negation, i.e., $NR\text{-Datalog}_s^{\neg A}$ programs such that predicates occurring in negated atoms cannot occur in rule heads.

Moreover, throughout this section we impose the further restriction that programs are such that *DL predicates do not occur in the head of rules*. We call *head-DL-free* the programs satisfying the above restriction. Such a restriction strengthens the lower bounds and undecidability results which are presented below.

Furthermore, we remark that we focus on *data complexity* of satisfiability, which in the framework of r-hybrid KBs ($\mathcal{H} = (\mathcal{K}, \mathcal{P})$ with $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$) 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} .

Finally, we point out that most of the proofs of the following theorems are obtained by exploiting and extending the proofs of recent results on query answering in DLs, in particular the results in [6, 29].

We start by analyzing r-hybrid KBs with $NR\text{-Datalog}$ programs.

Theorem 1. *Let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r-hybrid KB such that \mathcal{P} is a head-DL-free $NR\text{-Datalog}$ program. Then, under both FOL semantics and NM semantics:*

- when \mathcal{K} is either a $DL\text{-Lite}_{RDFS}$ KB or a $DL\text{-Lite}_R$ KB, deciding satisfiability of \mathcal{H} is in LOGSPACE with respect to data complexity;
- when \mathcal{K} is an \mathcal{EL} KB, deciding satisfiability of \mathcal{H} is PTIME-complete with respect to data complexity.

Proof (sketch). First, observe that by Proposition 2 FOL-satisfiability and NM-satisfiability coincide for the class of r-hybrid KBs considered. Then, for $DL\text{-Lite}_{RDFS}$ and $DL\text{-Lite}_R$ the thesis follows from the complexity results on answering unions of conjunctive queries in $DL\text{-Lite}_R$ [6] and from the fact that it is possible to reduce unsatisfiability of $(\mathcal{K}, \mathcal{P})$, where \mathcal{P} is nonrecursive, to the evaluation of unions of conjunctive queries over \mathcal{K} . In the case of \mathcal{EL} , the thesis follows from a similar argument and from the computational properties of answering unions of conjunctive queries in \mathcal{EL} [29, Theorem 4].

Then, we provide the following computational characterization of satisfiability in the presence of $NR\text{-Datalog}_s^\neq$ programs.

Theorem 2. *Let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r-hybrid KB such that \mathcal{P} is a head-DL-free $NR\text{-Datalog}_s^\neq$ program. Then, under both FOL semantics and NM semantics:*

- when \mathcal{K} is a $DL\text{-Lite}_{RDFS}$ KB, deciding satisfiability of \mathcal{H} is in LOGSPACE with respect to data complexity;
- when \mathcal{K} is an \mathcal{EL} KB, deciding satisfiability of \mathcal{H} is PTIME-complete with respect to data complexity;
- when \mathcal{K} is a $DL\text{-Lite}_R$ KB, deciding satisfiability of \mathcal{H} is NP-hard with respect to data complexity;

- when \mathcal{K} is an \mathcal{ALC} KB, satisfiability of \mathcal{H} is undecidable.

Proof (sketch). First, observe that by Proposition 2 FOL-satisfiability and NM-satisfiability coincide for the class of r-hybrid KBs considered. Then, for $DL-Lite_{RDFS}$ and \mathcal{EL} the thesis is a consequence of a property analogous to [29, Theorem 7], and to the data complexity of answering conjunctive queries in those DLs [6, 29], while for $DL-Lite_R$ the proof is by reduction from satisfiability of a 3-CNF propositional formula, in a way analogous to [29, Theorem 6]. Finally, in the case of \mathcal{ALC} the proof is by reduction from the unbounded tiling problem [4], in a way analogous to [29, Theorem 5].

We then analyze reasoning in r-hybrid KBs with $NR-Datalog^{\neq}$ programs.

Theorem 3. *Let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r-hybrid KB such that \mathcal{P} is a head-DL-free $NR-Datalog^{\neq}$ program. Then, under both FOL semantics and NM semantics:*

- when \mathcal{K} is a $DL-Lite_{RDFS}$ KB, deciding satisfiability of \mathcal{H} is in LOGSPACE with respect to data complexity;
- when \mathcal{K} is either a $DL-Lite_R$ KB or an \mathcal{EL} KB, satisfiability of \mathcal{H} is undecidable.

Proof (sketch). Again, we start by observing that by Proposition 2 FOL-satisfiability and NM-satisfiability coincide for the class of r-hybrid KBs considered. Then, for $DL-Lite_{RDFS}$ the proof is obtained by extending the result in [29, Theorem 11], while in the case of both $DL-Lite_R$ and \mathcal{EL} the proof is obtained by reducing the word problem for semigroups to satisfiability in such DLs, in a way analogous to Theorem 8 and Theorem 9 of [29].

Next, we are able to prove the following results for r-hybrid KBs with $NR-Datalog_s^{\neg}$ programs.

Theorem 4. *Let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r-hybrid KB such that \mathcal{P} is a head-DL-free $NR-Datalog_s^{\neg}$ program. Then, under both FOL semantics and NM semantics:*

- when \mathcal{K} is a $DL-Lite_{RDFS}$ KB, deciding satisfiability of \mathcal{H} is in LOGSPACE with respect to data complexity;
- when \mathcal{K} is an \mathcal{EL} KB, deciding satisfiability of \mathcal{H} is PTIME-complete with respect to data complexity;
- when \mathcal{K} is a $DL-Lite_R$ KB, deciding satisfiability of \mathcal{H} is NP-hard with respect to data complexity;
- when \mathcal{K} is an \mathcal{ALC} KB, satisfiability of \mathcal{H} is undecidable.

Proof (sketch). First, we consider the case of FOL-satisfiability. For $DL-Lite_{RDFS}$ and \mathcal{EL} the proof is obtained from [29, Theorem 14] and from the data complexity of answering conjunctive queries in those DLs [6, 29], while for $DL-Lite_R$ the proof is by reduction from satisfiability of a 3-CNF propositional formula, in a way analogous to [29, Theorem 13]. Finally, in the case of \mathcal{ALC} the proof is by reduction from the unbounded tiling problem [4], in a way analogous to [29, Theorem 12]. The above results

	$NR\text{-Datalog}$	$NR\text{-Datalog}_s^{\neq}$	$NR\text{-Datalog}^{\neq}$	$NR\text{-Datalog}_s^{-}$	$NR\text{-Datalog}^{-A}$	$NR\text{-Datalog}^{-}$
$DL\text{-Lite}_{RDFS}$	$\leq \text{LOGSPACE}$	$\leq \text{LOGSPACE}$	$\leq \text{LOGSPACE}$	$\leq \text{LOGSPACE}$	$= \text{NP}$	UNDEC.
$DL\text{-Lite}_R$	$\leq \text{LOGSPACE}$	$> \text{NP}$	UNDEC.	$> \text{NP}$	UNDEC.	UNDEC.
\mathcal{EL}	$= \text{PTIME}$	$= \text{PTIME}$	UNDEC.	$= \text{PTIME}$	UNDEC.	UNDEC.
from \mathcal{ALC} to \mathcal{SHIQ}	$= \text{NP}$	UNDEC.	UNDEC.	UNDEC.	UNDEC.	UNDEC.
\mathcal{DLR}	DECID., $\geq \text{NP}$	UNDEC.	UNDEC.	UNDEC.	UNDEC.	UNDEC.

Fig. 2. Decidability/data complexity of both FOL-satisfiability and NM-satisfiability in r-hybrid KBs (head-DL-free programs).

can be easily extended to the case of NM-satisfiability: in particular, the above reductions used for $DL\text{-Lite}_R$ and \mathcal{ALC} do not employ negated Datalog atoms in rules, hence by Proposition 2 such reductions also prove the thesis under the NM semantics.

Finally, we consider $NR\text{-Datalog}^{-A}$ programs, and provide the following results.

Theorem 5. *Let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r-hybrid KB such that \mathcal{P} is a head-DL-free $NR\text{-Datalog}^{-A}$ program. Then, under both FOL semantics and NM semantics:*

- when \mathcal{K} is a $DL\text{-Lite}_{RDFS}$ KB, deciding satisfiability of \mathcal{H} is NP-hard with respect to data complexity;
- when \mathcal{K} is either a $DL\text{-Lite}_R$ KB or an \mathcal{EL} KB, satisfiability of \mathcal{H} is undecidable.

Proof (sketch). First, we consider the case of FOL-satisfiability. For $DL\text{-Lite}_{RDFS}$ the proof is obtained from [29, Theorem 16], while for $DL\text{-Lite}_R$ and \mathcal{EL} the proof is by reduction from the unbounded tiling problem, in a way analogous to [29, Theorem 15]. Finally, the above reductions do not employ negated Datalog atoms in rules, hence by Proposition 2 such reductions also prove the thesis under the NM semantics.

The table displayed in Figure 2 summarizes the results presented in this section. In the table, each column corresponds to a different rule language, while each row corresponds to a different DL. Each cell reports the data complexity of satisfiability (both under FOL semantics and under NM semantics) in the corresponding combination of DL and rule language. If the problem is decidable, then hardness (\geq) and/or membership (\leq) and/or completeness ($=$) results are reported.

More precisely, observe that:

- the results for $NR\text{-Datalog}$ programs follow from Theorem 1 and from the results in [6, 13];
- the well-known translation of arbitrary first-order queries in $NR\text{-Datalog}^{-}$ allows for reducing satisfiability of first-order sentences to satisfiability of r-hybrid KBs with $NR\text{-Datalog}^{-}$ programs for any choice of the DL language, which immediately implies undecidability of reasoning in this class of r-hybrid KBs.

Finally, due to the correspondence between unsatisfiability and entailment in r-hybrid KBs illustrated in Section 3 (Proposition 1), it is also immediate to turn these results (obtained for satisfiability of programs with constraints) into results for skeptical entailment (also for classes of programs without constraints).

5 Results for weakly DL-safe rules

In this section we consider the weakly DL-safe integration of DLs and disjunctive Datalog. More precisely, we extend the weak DL-safeness restriction defined in the framework of $\mathcal{DL}+log$ [28] to the r-hybrid KBs defined in Section 3, thus extending the setting presented in [28] by considering the presence of inequality and of negation of DL predicates. Then, we extend the computational results presented in [28] to such a class of r-hybrid KBs.

Weak DL-safeness is formally defined as follows.

Definition 2. *Given a r-hybrid KB $\mathcal{H} = (\mathcal{K}, \mathcal{P})$, we say that \mathcal{P} is weakly DL-safe if every rule R in \mathcal{P} of the form (1) is such that, for every variable x appearing in R , either x occurs in a positive Datalog atom in the body of R , or x only occurs in positive DL atoms in the body of R .*

In other words, weak DL-safeness imposes (besides the usual Datalog safeness) the following condition: every variable that is either a head variable or a variable occurring in a negated atom or in an inequality occurs in a positive Datalog atom in the body of the rule. Such a restriction only constrains the interaction between the DL KB and the Datalog program, in the sense that neither it imposes any additional restriction on the rules if the DL KB is empty, nor it imposes any restriction on the DL KB.

We now show decidability of reasoning in r-hybrid KBs under the above restriction. To this aim, we start from the algorithm for NM-satisfiability in $\mathcal{DL}+log$ presented in [28] and extend it to the broader class of rules considered here. Due to space limits, we do not report details on the algorithm, which is actually very similar to the one reported in [28]. Such an algorithm checks satisfiability of a r-hybrid KB by solving a finite number of *Boolean CQ/UCQ containment* problems in DLs. Boolean CQ/UCQ containment is the problem of checking the containment between two queries q_1 and q_2 with respect to a DL KB \mathcal{K} , where q_1 is a Boolean conjunctive query and q_2 is a Boolean union of conjunctive queries (this problem is also known as *existential entailment* [20]).

Based on such an algorithm, we are able to extend the general decidability result of [28] to the present class of r-hybrid KBs. Formally:

Theorem 6. *Let \mathcal{DL} be a description logic and let $\mathcal{H} = (\mathcal{K}, \mathcal{P})$ be a r-hybrid KB, where \mathcal{K} is a \mathcal{DL} KB and \mathcal{P} is a weakly DL-safe Datalog^{-v} program. NM-satisfiability (as well as FOL-satisfiability) of \mathcal{H} is decidable iff Boolean CQ/UCQ containment is decidable in \mathcal{DL} .*

In particular, the above theorem and the results on CQ/UCQ containment in DLs presented in [20, 7, 15, 25] imply the following property: *for all the DLs studied in this paper, NM-satisfiability (as well as FOL-satisfiability) of weakly DL-safe r-hybrid KBs is decidable.*

Moreover, based on the above cited results and on our technique for NM-satisfiability, we are able to provide a computational characterization of r-hybrid KBs with weakly DL-safe rules for all the DLs and all the classes of nonrecursive programs above considered. More specifically, the table in Figure 3 summarizes the results on data complexity of NM-satisfiability (as well as for FOL-satisfiability) which hold for

	$NR\text{-Datalog}, NR\text{-Datalog}^{\neq}, NR\text{-Datalog}^{\neg}$
$DL\text{-Lite}_{RDFS}$	$\leq \text{LOGSPACE}$
$DL\text{-Lite}_R$	$\leq \text{LOGSPACE}$
\mathcal{EL}	$= \text{PTIME}$
from \mathcal{ALC} to \mathcal{SHIQ}	$= \text{NP}$
\mathcal{DLR}	$\text{DECIDABLE}, \geq \text{NP}$

Fig. 3. Data complexity of both NM-satisfiability and FOL-satisfiability in r-hybrid KBs with nonrecursive weakly DL-safe programs.

the class of r-hybrid KBs with weakly DL-safe rules. The complexity is the same for all the classes of nonrecursive Datalog rules considered in this paper.

A comparison of the table in Figure 3 with the previous one in Figure 2 allows us to evaluate the impact of the weak-DL-safeness assumption on the complexity of reasoning in r-hybrid KBs. Indeed, restricting the interaction between DLs and rules through the weak DL-safeness condition allows for using even very expressive DLs as the ontology language of the r-hybrid KB, without losing decidability of reasoning. In particular, Theorem 6 implies that, under the weak DL-safeness condition, it is possible to combine every DL considered in this paper with full Datalog^{¬∨} programs (i.e., with recursion, inequality, negation, and disjunction in the head), and obtain a decidable formalism. Moreover, Figure 3 shows that, for all the DLs and the classes of nonrecursive Datalog rules considered in this paper, when we impose weak DL-safeness the data complexity of reasoning is no worse than the data complexity of reasoning in the absence of rules: i.e., adding weakly DL-safe nonrecursive rules does not actually affect data complexity of reasoning in all the DLs considered.

On the other hand, the unrestricted integration of DLs and rules imposes severe restrictions on the expressive power of both the DL component and the rule component: indeed, as explicitly shown by Figure 2, decidability in the presence of inequality or negation in rules can be regained at the price of restricting both the ontology language to DLs of very little expressiveness and the rule language to extremely limited fragments of Datalog.

6 Conclusions

In this paper we have tried to extend the computational analysis of reasoning in systems integrating Description Logics ontologies and Datalog rules. To this aim, we have considered a group of Description Logics which, from the viewpoint of the expressive power, lie within the range from RDFS to OWL, and thus constitute very important classes of ontology formalisms with respect to Semantic Web applications. Moreover, we have considered disjunctive Datalog and several subclasses of it, with special emphasis on nonrecursive and nondisjunctive fragments.

In our opinion, the results presented in Section 4 clearly show that the unrestricted interaction of DLs and Datalog is computationally very hard even in the absence of recursion in rules. This contrasts with the general opinion that recursion is a necessary

feature for rules to rise the undecidability issue in their integration with DL ontologies. So, surprisingly, the various “safeness” restrictions which have been defined to regain decidability in the interaction between DLs and recursive Datalog, appear as necessary restrictions even when rules are not recursive. In this respect, the results in Section 5 further enlarge the class of Description Logics and rules with decidable, restricted integration, and provide a refined computational analysis for the integration of weakly DL-safe rules with the Description Logics considered in this paper.

The present study can be extended in several directions. In our opinion, the most interesting ones are the following:

- the analysis presented in Section 4 should be extended to other very promising tractable DLs recently defined, in particular *HornSHIQ* [19], \mathcal{EL}^{++} [2] and *DL-Lite_F* [5];
- the analysis presented in Section 4 should be further extended to classes of disjunctive programs;
- it would be very interesting, for the decidable cases of Figure 2, to provide upper bounds for *non-head-DL-free* programs;
- with respect to the results presented in Section 5, an important open issue is whether it is possible to identify other forms of decidable interaction between DL-KBs and rules, which overcome the expressive limitations of the weak DL-safeness (see [28]). An approach in this direction is presented in [22], which is based on the use of a modal autoepistemic logic, as well as the approach in [8]. Moreover, other interesting approaches have been presented. Some of the most recent ones study the combination of DLs and rules under a different semantic approach [21] or under different restrictions on variables in rules [17].

Acknowledgments This research has been partially supported by FET project TONES (Thinking ONtologiES), funded by the EU under contract number FP6-7603, 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).

References

1. <http://www.w3.org/TR/rdf-schema/>.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the \mathcal{EL} envelope. In *Proc. of IJCAI 2005*, pages 364–369, 2005.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
4. R. Berger. The undecidability of the domino problem. *Mem. Amer. Math. Soc.*, 66:1–72, 1966.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, pages 602–607, 2005.
6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR 2006*, 2006.

7. D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS'98*, pages 149–158, 1998.
8. J. de Bruijn, T. Eiter, A. Polleres, and H. Tompits. Embedding non-ground logic programs into autoepistemic logic for knowledge-base combination. In *Proc. of IJCAI 2007*, 2007.
9. T. Eiter, G. Gottlob, and H. Mannilla. Disjunctive Datalog. *ACM Trans. on Database Systems*, 22(3):364–418, 1997.
10. T. Eiter, G. Ianni, R. Schindlauer, and H. Tompits. Effective integration of declarative rules with external evaluations for semantic-web reasoning. In *Proc. of ESWC 2006*, pages 273–287, 2006.
11. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. In *Proc. of KR 2004*, pages 141–151, 2004.
12. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
13. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic SHIQ. In *Proc. of IJCAI 2007*, pages 399–404, 2007.
14. B. Glimm, I. Horrocks, and U. Sattler. Conjunctive query answering for description logics with transitive roles. In *Proc. of DL 2006*. CEUR Electronic Workshop Proceedings, <http://ceur-ws.org/Vol-189>, 2006.
15. B. Glimm, I. Horrocks, and U. Sattler. Conjunctive query answering for the description logic SHOIQ. Technical report, University of Manchester, UK, 2006.
16. B. C. Grau. A possible simplification of the semantic web architecture. In *Proc. of the 13th Int. World Wide Web Conf. (WWW 2004)*, pages 704–713, 2004.
17. S. Heymans, J. de Bruijn, L. Predoiu, C. Feier, and D. V. Nieuwenborgh. Guarded hybrid knowledge bases. *Theory and Practice of Logic Programming*, 8(3):411–429, 2008.
18. I. Horrocks and P. F. Patel-Schneider. A proposal for an OWL rules language. In *Proc. of the 13th Int. World Wide Web Conf. (WWW 2004)*, pages 723–731, 2004.
19. U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. of IJCAI 2005*, pages 466–471, 2005.
20. A. Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104(1–2):165–209, 1998.
21. T. Lukasiewicz. A novel combination of answer set programming with description logics for the semantic web. In *Proc. of ESWC 2007*, pages 384–398, 2007.
22. B. Motik and R. Rosati. A faithful integration of description logics with logic programming. In *Proc. of IJCAI 2007*, 2007. To appear.
23. B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. *J. of Web Semantics*, 3(1):41–60, 2005.
24. M. M. Ortiz, D. Calvanese, and T. Eiter. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proc. of AAAI 2006*, 2006.
25. M. M. Ortiz de la Fuente, D. Calvanese, and T. Eiter. Data complexity of answering unions of conjunctive queries in SHIQ. Technical report, Fac. of Computer Science, Free Univ. of Bozen-Bolzano, Mar. 2006. Available at <http://www.inf.unibz.it/calvanese/papers/orti-calv-eite-TR-2006-03.pdf>.
26. P. F. Patel-Schneider, P. J. Hayes, I. Horrocks, and F. van Harmelen. OWL web ontology language; semantics and abstract syntax. W3C candidate recommendation, <http://www.w3.org/tr/owl-semantics/>, november 2002.
27. R. Rosati. On the decidability and complexity of integrating ontologies and rules. *J. of Web Semantics*, 3(1):61–73, 2005.
28. R. Rosati. DL+log: tight integration of Description Logics and disjunctive Datalog. In *Proc. of KR 2006*, pages 68–78, 2006.
29. R. Rosati. The limits of querying ontologies. In *Proc. of ICDT 2007*, pages 164–178, 2007.