Combining Logic Programming with Description Logics and Machine Learning for the Semantic Web

Francesca A. Lisi and Floriana Esposito

Dipartimento di Informatica, Università degli Studi di Bari
Via E. Orabona 4, 70125 Bari, Italy
{lisi, esposito}@di.uniba.it

Abstract. In this paper we consider an extension of Logic Programming that tackles the Semantic Web challenge of acquiring rules combined with ontologies. To face this bottleneck problem we propose a framework that resorts to the expressive and deductive power of $\mathcal{DL}$+log and adopts the methodological apparatus of Inductive Logic Programming.

1 Introduction

Combining rules and ontologies is a hot topic in the (Semantic) Web area as testified by the intense activity and the standardization efforts of the Rule Interchange Format working group at W3C. Yet the debate around a unified language for (Semantic) Web rules is still open. Indeed, combining rules and ontologies raises several issues in Knowledge Representation (KR) due to the many differences between the underlying logics, Clausal Logics (CLs) [17] and Description Logics (DLs) [1] respectively. Among the many recent KR proposals, $\mathcal{DL}$+log [23] is a very powerful framework that allows for the tight integration of DLs and disjunctive Datalog with negation (Datalog$^{\neg\vee}$) [7]. A point in favour of $\mathcal{DL}$+log is its decidability for many DLs, notably for $\mathcal{SHIQ}$ [12]. Since the design of OWL has been based on the $\mathcal{SH}$ family of very expressive DLs [11], $\mathcal{SHIQ}$+log is a good candidate for investigation in the Semantic Web context.

The upcoming standard rule language for the Semantic Web, if well-founded from the KR viewpoint, will be equipped with reasoning algorithms. In KR tradition deductive reasoning is the most widely studied. Yet, other forms of reasoning will become necessary. E.g., acquiring and maintaining Semantic Web rules is very demanding and can be automated though partially by applying Machine Learning algorithms. In this paper, we consider a decidable instantiation of $\mathcal{DL}$+log obtained by choosing $\mathcal{SHIQ}$ for the DL part and Datalog$^{\neg}$ for the CL part, and face the problem of defining inductive reasoning mechanisms on it. To solve the problem, we propose to resort to the methodological apparatus of that form of Machine Learning known under the name of Inductive Logic Programming (ILP) [19]. We extend some known ILP techniques to $\mathcal{SHIQ}+\text{LOG}^{\neg}$ and illustrate them with examples relevant to the Semantic Web context.

The paper is organized as follows. Section 2 briefly introduces hybrid DL-CL formalisms and ILP. Section 3 introduces the KR framework of $\mathcal{DL}$+log. Section 4 defines the ILP framework for inducing $\mathcal{SHIQ}$+log$^{\neg}$ rules. Section 5 provides a comparative analysis of our proposal with related work. Section 6 concludes the paper with final remarks.

2 Background

2.1 Logic Programming and Description Logics

Description Logics (DLs) are a family of KR formalims that allow for the specification of knowledge in terms of classes (*concepts*), binary relations between classes (*roles*), and instances (*individuals*) [1]. Complex concepts can be defined from atomic concepts and roles by means of constructors (see Table 1). E.g., concept descriptions in the basic DL $\mathcal{AL}$ are formed according to only the constructors of atomic negation, concept conjunction, value restriction, and limited existential restriction. The DLs $\mathcal{ALC}$ and $\mathcal{ALN}$ are members of the $\mathcal{AL}$ family. The former extends $\mathcal{AL}$ with (arbitrary) concept negation (or complement), whereas the latter with number restriction. The DL $\mathcal{ALCNR}$ adds to the constructors inherited from $\mathcal{ALC}$ and $\mathcal{ALN}$ a further one: role intersection (see Table 1). Conversely, in the DL $\mathcal{SHIQ}$ [12] it is allowed to invert roles and to express qualified number restrictions of the form $\geq nS.C$ and $\leq nS.C$ where S is a simple role (see Table 1).

A DL knowledge base (KB) can state both is-a relations between concepts (*axioms*) and instance-of relations between individuals (resp. couples of individuals) and concepts (resp. roles) (*assertions*). Concepts and axioms form the TBox whereas individuals and assertions form the ABox. A $\mathcal{SHIQ}$ KB encompasses also a RBox which consists of axioms concerning abstract roles. The semantics of DLs is usually defined through a mapping to First Order Logic (FOL) [2]. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for a DL KB consists of a non-empty domain $\Delta^{\mathcal{I}}$ and a mapping function $\cdot^{\mathcal{I}}$. In particular, individuals are mapped to elements of $\Delta^{\mathcal{I}}$ such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (*Unique Names Assumption* (UNA) [21]). Yet in $\mathcal{SHIQ}$ UNA does not hold by default [10]. Thus individual equality (inequality) assertions may appear in a $\mathcal{SHIQ}$ KB (see Table 1). Also the KB represents many different interpretations, i.e. all its models. This is coherent with the *Open World Assumption* (OWA) that holds in FOL semantics. The main reasoning task for a DL KB is the *consistency check* that is performed by applying decision procedures based on tableau calculus. Decidability of reasoning is crucial in DLs.

The integration of DLs and Logic Programming follows the tradition of KR research on *hybrid systems*, i.e. those systems which are constituted by two or more subsystems dealing with distinct portions of a single KB by performing specific reasoning procedures [8], and gives raise to KR systems that will be referred to as DL-CL hybrid systems in the rest of the paper. The motivation for investigating and developing such systems is to improve on *representational adequacy* and *deductive power* by preserving *decidability*. In particular, combining DLs with CLs can easily yield to undecidability if the interface between

Table 1. Syntax and semantics of DLs.

bottom (resp. top) concept	$\bot$ (resp. $\top$)	$\emptyset$ (resp. $\Delta^{\mathcal{I}}$)
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
(abstract) role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
(abstract) inverse role	R^{-}	$(R^{\mathcal{I}})^{-}$
(abstract) individual	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept intersection	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
concept union	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
value restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y\ (x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y\ (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
at least number restriction	$\geq nR$	$\{x \in \Delta^{\mathcal{I}} \mid \|\{y\|(x, y) \in R^{\mathcal{I}}\}\| \geq n\}$
at most number restriction	$\leq nR$	$\{x \in \Delta^{\mathcal{I}} \mid \|\{y\|(x, y) \in R^{\mathcal{I}}\}\| \leq n\}$
at least qualif. number restriction	$\geq nS.C$	$\{x \in \Delta^{\mathcal{I}} \mid \|\{y \in C^{\mathcal{I}}\|(x, y) \in S^{\mathcal{I}}\}\| \geq n\}$
at most qualif. number restriction	$\leq nS.C$	$\{x \in \Delta^{\mathcal{I}} \mid \|\{y \in C^{\mathcal{I}}\|(x, y) \in S^{\mathcal{I}}\}\| \leq n\}$
role intersection	$R_1 \sqcap R_2$	$R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}}$
concept equivalence axiom	$C_1 \equiv C_2$	$C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$
concept subsumption axiom	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
role equivalence axiom	$R \equiv S$	$R^{\mathcal{I}} = S^{\mathcal{I}}$
role inclusion axiom	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
concept assertion	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$\langle a, b\rangle : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
individual equality assertion	$a \approx b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
individual inequality assertion	$a \not\approx b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

them is not reduced. In [14] the family Carin of languages combining any DL and HCL is presented. Among the many important results of this study, it is proved that query answering in a logic obtained by extending $\mathcal{ALCNR}$ with non-recursive Datalog rules, where both concepts and roles can occur in rule bodies, is decidable. Query answering is decided using *constrained SLD-resolution*, i.e. an extension of SLD-resolution with a modified version of tableau calculus. Another DL-CL hybrid system is $\mathcal{AL}$-log [6] that integrates $\mathcal{ALC}$ [26] and Datalog [4] by constraining the variables occurring in the body of rules with $\mathcal{ALC}$ concept assertions. Constrained SLD-resolution for $\mathcal{AL}$-log is decidable and offers a complete and sound method for answering ground queries by refutation. Besides decidability, another relevant issue is *DL-safeness* of hybrid DL-CL systems [22]. A safe interaction between the DL and the CL part of an hybrid KB allows to solve the semantic mismatch between DLs and CLs due to the different inferences that can be made under OWA and CWA respectively. In this respect, $\mathcal{AL}$-log is DL-safe whereas Carin is not.

2.2 Logic Programming and Machine Learning

The research area born at the intersection of Logic Programming and Machine Learning, more precisely Concept Learning [18], is known under the name of Inductive Logic Programming (ILP) [19]. From Logic Programming ILP has borrowed the KR framework, i.e. Horn Clausal Logic (HCL). From Concept Learning it has inherited the inferential mechanisms for induction, the most prominent of which is *generalization* characterized as search through a partially ordered space of hypotheses. According to this vision, in ILP a hypothesis is a clausal theory (i.e., a set of rules) and the induction of a single clause (rule) requires (i) structuring, (ii) searching and (iii) bounding the space of hypotheses. First we focus on (i) by clarifying the notion of *ordering* for clauses. An ordering allows for determining which one, between two clauses, is more general than the other. Actually quasi-orders are considered, therefore uncomparable pairs of clauses are admitted. One such ordering is *θ-subsumption* [20]: Given two clauses C and D, we say that C θ-subsumes D if there exists a substitution θ, such that $C\theta \subseteq D$. Given the usefulness of Background Knowledge (BK) in ILP, orders have been proposed that reckon with it, e.g. Buntine's *generalized subsumption* [3]. Generalized subsumption only applies to definite clauses and the BK should be a definite program. Once structured, the space of hypotheses can be searched (ii) by means of refinement operators. A *refinement operator* is a function which computes a set of specializations or generalizations of a clause according to whether a top-down or a bottom-up search is performed. The two kinds of refinement operator have been therefore called *downward* and *upward*, respectively. The definition of refinement operators presupposes the investigation of the properties of the various quasi-orders and is usually coupled with the specification of a declarative bias for bounding the space of clauses (iii). *Bias* concerns anything which constrains the search for theories, e.g. a *language bias* specifies syntactic constraints on the clauses in the search space.

Induction with ILP generalizes from individual instances/observations in the presence of BK, finding *valid hypotheses*. Validity depends on the underlying *setting*. At present, there exist several formalizations of induction in clausal logic that can be classified according to the following two orthogonal dimensions: the *scope of induction* (discrimination vs characterization) and the *representation of observations* (ground definite clauses vs ground unit clauses) [5]. *Discriminant induction* aims at inducing hypotheses with discriminant power as required in tasks such as classification. In classification, observations encompass both positive and negative examples. *Characteristic induction* is more suitable for finding regularities in a data set. This corresponds to learning from positive examples only. The second dimension affects the notion of *coverage*, i.e. the condition under which a hypothesis explains an observation. In *learning from entailment* (or *from implications*), hypotheses are clausal theories, observations are ground definite clauses, and a hypothesis covers an observation if the hypothesis logically entails the observation. In *learning from interpretations*, hypotheses are clausal theories, observations are Herbrand interpretations (ground unit clauses) and a hypothesis covers an observation if the observation is a model for the hypothesis.

3 Combining LP and DLs with $\mathcal{DL}$+log

The KR framework of $\mathcal{DL}$+log [23] allows for the tight integration of DLs [1] and DATALOG$^{\neg\vee}$ [7]. More precisely, it allows a DL KB to be extended with *weakly-safe* DATALOG$^{\neg\vee}$ rules. The condition of weak safeness allows to overcome the main representational limits of the approaches based on the DL-safeness condition, e.g. the possibility of expressing conjunctive queries (CQ) and unions of conjunctive queries (UCQ)[1], by keeping the integration scheme still decidable. To a certain extent, $\mathcal{DL}$+log is between $\mathcal{AL}$-log [6] and CARIN [14].

3.1 Syntax

Formulas in $\mathcal{DL}$+log are built upon three mutually disjoint predicate alphabets: an alphabet of concept names P_C, an alphabet of role names P_R, and an alphabet of DATALOG predicates P_D. We call a predicate p a *DL-predicate* if either $p \in P_C$ or $p \in P_R$. 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 $p \in P_C \cup P_R$, the atom is called a *DL-atom*, while if $p \in P_D$, it is called a DATALOG *atom*.

Given a description logic $\mathcal{DL}$, a $\mathcal{DL}$+log KB $\mathcal{B}$ is a pair (Σ, Π), where Σ is a $\mathcal{DL}$ KB and Π is a set of DATALOG$^{\neg\vee}$ rules, where each rule R has the form

$$p_1(\boldsymbol{X_1}) \vee \ldots \vee p_n(\boldsymbol{X_n}) \leftarrow$$
$$r_1(\boldsymbol{Y_1}), \ldots, r_m(\boldsymbol{Y_m}), s_1(\boldsymbol{Z_1}), \ldots, s_k(\boldsymbol{Z_k}), \neg u_1(\boldsymbol{W_1}), \ldots, \neg u_h(\boldsymbol{W_h})$$

with $n, m, k, h \geq 0$, each $p_i(\boldsymbol{X_i})$, $r_j(\boldsymbol{Y_j})$, $s_l(\boldsymbol{Z_l})$, $u_k(\boldsymbol{W_k})$ is an atom and:

- each p_i is either a DL-predicate or a DATALOG predicate;
- each r_j, u_k is a DATALOG predicate;
- each s_l is a DL-predicate;
- (DATALOG safeness) every variable occurring in R must appear in at least one of the atoms $r_1(\boldsymbol{Y_1}), \ldots, r_m(\boldsymbol{Y_m}), s_1(\boldsymbol{Z_1}), \ldots, s_k(\boldsymbol{Z_k})$;
- (weak safeness) every head variable of R must appear in at least one of the atoms $r_1(\boldsymbol{Y_1}), \ldots, r_m(\boldsymbol{Y_m})$.

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 can be expressed as follows: every variable of R must appear in at least one of the atoms $r_1(\boldsymbol{Y_1}), \ldots, r_m(\boldsymbol{Y_m})$. 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, we can assume that in a $\mathcal{DL}$+log KB (Σ, Π) all constants occurring in Σ also occur in Π.

[1] A *Boolean UCQ* over a predicate alphabet P is a first-order sentence of the form $\exists \boldsymbol{X}.conj_1(\boldsymbol{X}) \vee \ldots \vee conj_n(\boldsymbol{X})$, where $\boldsymbol{X}$ is a tuple of variable symbols and each $conj_i(\boldsymbol{X})$ is a set of atoms whose predicates are in P and whose arguments are either constants or variables from $\boldsymbol{X}$. A *Boolean CQ* corresponds to a Boolean UCQ in the case when $n = 1$.

Example 1. Let us consider a $\mathcal{DL}$+log KB $\mathcal{B}$ (adapted from [23]) integrating the following DL-KB Σ (ontology about persons)

$[A1]$ `PERSON` $\sqsubseteq \exists$ `FATHER`$^{-}$`.MALE`
$[A2]$ `MALE` $\sqsubseteq$ `PERSON`
$[A3]$ `FEMALE` $\sqsubseteq$ `PERSON`
$[A4]$ `FEMALE` $\sqsubseteq \neg$`MALE`
`MALE(Bob)`
`PERSON(Mary)`
`PERSON(Paul)`
`FATHER(John,Paul)`

and the following DATALOG$^{\neg\vee}$ program Π (rules about students):

$[R1]$ `boy(X)` $\leftarrow$ `enrolled(X,c1,bsc)`, `PERSON(X)`, $\neg$`girl(X)`
$[R2]$ `girl(X)` $\leftarrow$ `enrolled(X,c2,msc)`, `PERSON(X)`
$[R3]$ `boy(X)`$\vee$ `girl(X)` $\leftarrow$ `enrolled(X,c3,phd)`, `PERSON(X)`
$[R4]$ `FEMALE(X)` $\leftarrow$ `girl(X)`
$[R5]$ `MALE(X)` $\leftarrow$ `boy(X)`
$[R6]$ `man(X)` $\leftarrow$ `enrolled(X,c3,phd)`, `FATHER(X,Y)`
`enrolled(Paul,c1,bsc)`
`enrolled(Mary,c1,bsc)`
`enrolled(Mary,c2,msc)`
`enrolled(Bob,c3,phd)`
`enrolled(John,c3,phd)`

Note that the rules mix DL-literals and DATALOG-literals. Notice that the variable `Y` in rule $R6$ is weakly-safe but not DL-safe, since `Y` does not occur in any DATALOG predicate in $R6$.

3.2 Semantics

For $\mathcal{DL}$+log two semantics have been defined: a first-order logic (FOL) semantics and a nonmonotonic (NM) semantics. In particular, the latter extends the stable model semantics of DATALOG$^{\neg\vee}$ [9]. According to it, DL-predicates are still interpreted under OWA, while DATALOG predicates are interpreted under CWA. Notice that, under both semantics, entailment can be reduced to satisfiability. In a similar way, it can be seen that CQ answering can be reduced to satisfiability in $\mathcal{DL}$+log. Consequently, Rosati [23] concentrates on the satisfiability problem in $\mathcal{DL}$+log KBs. It has been shown that, when the rules are positive disjunctive, the above two semantics are equivalent with respect to the satisfiability problem. In particular, FOL-satisfiability can always be reduced (in linear time) to NM-satisfiability. Hence, the satisfiability problem under the NM semantics is in the focus of interest.

Example 2. With reference to Example 1, 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 Σ);
– `MALE(Paul)` (due to rule R5);
– `FEMALE(Mary)` (due to rule R4).

Notice that $\mathcal{B} \models_{NM}$`FEMALE(Mary)`, while $\Sigma \not\models_{FOL}$ `FEMALE(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 FOL semantics of $\mathcal{DL}$+log-KBs, since it can be verified that $\mathcal{B} \models_{FOL}$`FEMALE(Mary)` in this case.

3.3 Reasoning

The problem statement of satisfiability for finite $\mathcal{DL}$+log KBs relies on the following problem known as the *Boolean CQ/UCQ containment problem*[2] in DLs: Given a $\mathcal{DL}$-TBox $\mathcal{T}$, a Boolean CQ Q_1 and a Boolean UCQ Q_2 over the alphabet $P_C \cup P_R$, 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 $\mathcal{I}$. The algorithm NMSAT-$\mathcal{DL}$+log for deciding NM-satisfiability of $\mathcal{DL}$+log KBs looks for a guess (G_P, G_N) of the Boolean CQs in the DL-grounding of Π, denoted as $gr_p(\Pi)$, that is consistent with the $\mathcal{DL}$-KB Σ (Boolean CQ/UCQ containment problem) and such that the Datalog$^{\neg\vee}$ program $\Pi(G_P, G_N)$ has a stable model. Details of how obtaining $gr_p(\Pi)$ and $\Pi(G_P, G_N)$ can be found in [23].

The decidability of reasoning in $\mathcal{DL}$+log, thus of ground query answering, depends on the decidability of the Boolean CQ/UCQ containment problem in $\mathcal{DL}$. Consequently, ground queries can be answered by applying NMSAT-$\mathcal{DL}$+log.

Theorem 1 *[23] 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 $\mathcal{DL}$.*

Corollary 1. *Given a $\mathcal{DL}$+log KB (Σ, Π) and a ground atom α, $(\Sigma, \Pi) \models \alpha$ iff $(\Sigma, \Pi \cup \{\leftarrow \alpha\})$ is unsatisfiable.*

From Theorem 1 and from previous results on query answering and query containment in DLs, it follows the decidability of reasoning in several instantiations of $\mathcal{DL}$+log. Since $\mathcal{SHIQ}$ is the most expressive DL for which the Boolean CQ/UCQ containment is decidable [10], we consider $\mathcal{SHIQ}$+log$^{\neg}$ (i.e. $\mathcal{SHIQ}$ extended with weakly-safe Datalog$^{\neg}$ rules) as the KR framework in our study of ILP for the Semantic Web.

[2] This problem was called *existential entailment* in [14].

4 Inducing $\mathcal{SHIQ}$+log$^\neg$ Rules with ILP

We consider the task of inducing new $\mathcal{SHIQ}$+log$^\neg$ rules from an already existing $\mathcal{SHIQ}$+log$^\neg$ KB. At this stage of work the scope of induction does not matter. Therefore the term 'observation' is to be preferred to the term 'example'. We choose to work within the setting of *learning from interpretations* which requires an observation to be represented as a set of ground unit clauses.

We assume that the data are represented as a $\mathcal{SHIQ}$+log$^\neg$ KB $\mathcal{B}$ where the intensional part $\mathcal{K}$ (i.e., the TBox $\mathcal{T}$ plus the set Π_R of rules) plays the role of *background knowledge* and the extensional part (i.e., the ABox $\mathcal{A}$ plus the set Π_F of facts) contributes to the definition of *observations*. Therefore ontologies may appear as input to the learning problem of interest.

Example 3. Suppose we have a $\mathcal{SHIQ}$+log$^\neg$ KB (adapted from [23]) consisting of the following intensional knowledge $\mathcal{K}$:

$[A1]$ $\texttt{RICH}\sqcap\texttt{UNMARRIED} \sqsubseteq \exists\ \texttt{WANTS-TO-MARRY}^{-}.\top$
$[R1]$ $\texttt{RICH(X)} \leftarrow \texttt{famous(X)}, \neg\ \texttt{scientist(X)}$

and the following extensional knowledge $\mathcal{F}$:

UNMARRIED(Mary)
UNMARRIED(Joe)
famous(Mary)
famous(Paul)
famous(Joe)
scientist(Joe)

that can be split into $\mathcal{F}_{\texttt{Joe}} = \{\texttt{UNMARRIED(Joe)}, \texttt{famous(Joe)}, \texttt{scientist(Joe)}\}$, $\mathcal{F}_{\texttt{Mary}} = \{\texttt{UNMARRIED(Mary)}, \texttt{famous(Mary)}\}$, and $\mathcal{F}_{\texttt{Paul}} = \{\texttt{famous(Paul)}\}$.

The language $\mathcal{L}$ of hypotheses must allow for the generation of $\mathcal{SHIQ}$+log$^\neg$ rules starting from three disjoint alphabets $P_C(\mathcal{L}) \subseteq P_C(\mathcal{B})$, $P_R(\mathcal{L}) \subseteq P_R(\mathcal{B})$, and $P_D(\mathcal{L}) \subseteq P_D(\mathcal{B})$. More precisely, we consider linked[3] and range-restricted[4] weakly-safe Datalog$^\neg$ clauses of the form

$$p(\boldsymbol{X}) \leftarrow r_1(\boldsymbol{Y_1}), \ldots, r_m(\boldsymbol{Y_m}), s_1(\boldsymbol{Z_1}), \ldots, s_k(\boldsymbol{Z_k}), \neg u_1(\boldsymbol{W_1}), \ldots, \neg u_h(\boldsymbol{W_h})$$

where the unique literal $p(\boldsymbol{X})$ in the head represents the target predicate, denoted as c if p is a Datalog-predicate and as C if p is a $\mathcal{SHIQ}$-predicate. In the following we provide examples for these two cases of rule learning, one aimed at inducing $c(\boldsymbol{X}) \leftarrow$ rules and the other $C(\boldsymbol{X}) \leftarrow$ rules. The former kind of rule will enrich the Datalog part of the KB, whereas the latter will extend the DL part (i.e., the input ontology).

[3] A clause H is *linked* if each literal $l_i \in H$ is linked. A literal $l_i \in H$ is linked if at least one of its terms is linked. A term t in some literal $l_i \in H$ is linked with linking-chain of length 0, if t occurs in $head(H)$, and with linking-chain of length $d+1$, if some other term in l_i is linked with linking-chain of length d. The link-depth of a term t in l_i is the length of the shortest linking-chain of t.

[4] A clause H is *range-restricted* if each variable occurring in $head(H)$ also occur in $body(H)$.

Example 4. Suppose that the DATALOG-predicate happy is the target predicate and the set $P_D(\mathcal{L}^{\texttt{happy}}) \cup P_C(\mathcal{L}^{\texttt{happy}}) \cup P_R(\mathcal{L}^{\texttt{happy}}) = \{\texttt{famous/1}\} \cup \{\texttt{RICH/1}\} \cup \{\texttt{WANTS-TO-MARRY/2}\}$ provides the building blocks for the language $\mathcal{L}^{\texttt{happy}}$. The following $\mathcal{SHIQ}$+log$^{\neg}$ rules

$H_1^{\texttt{happy}}$ `happy(X) ← RICH(X)`
$H_2^{\texttt{happy}}$ `happy(X) ← famous(X)`
$H_3^{\texttt{happy}}$ `happy(X) ← famous(X), WANTS-TO-MARRY(Y,X)`

belonging to $\mathcal{L}^{\texttt{happy}}$ can be considered hypotheses for the target predicate happy. Note that $H_3^{\texttt{happy}}$ is weakly-safe.

Example 5. Suppose now that the target predicate is the DL-predicate LONER. If $\mathcal{L}^{\texttt{LONER}}$ is defined over $P_D(\mathcal{L}^{\texttt{LONER}}) \cup P_C(\mathcal{L}^{\texttt{LONER}}) = \{\texttt{famous/1}, \texttt{scientist/1}\} \cup \{\texttt{UNMARRIED/1}\}$, then the following $\mathcal{SHIQ}$+log$^{\neg}$ rules

$H_1^{\texttt{LONER}}$ `LONER(X) ← scientist(X)`
$H_2^{\texttt{LONER}}$ `LONER(X) ← scientist(X), UNMARRIED(X)`
$H_3^{\texttt{LONER}}$ `LONER(X) ← ¬famous(X)`

belong to $\mathcal{L}^{\texttt{LONER}}$ and represent hypotheses for the target predicate LONER.

In order to support with ILP techniques the induction of $\mathcal{SHIQ}$+log$^{\neg}$ rules, the language $\mathcal{L}$ of hypotheses needs to be equipped with a generality order $\succeq$, and a coverage relation *covers* so that $(\mathcal{L}, \succeq)$ is a search space and *covers* defines the mappings from $(\mathcal{L}, \succeq)$ to the set O of observations. The next subsections are devoted to these issues.

4.1 The hypothesis ordering

The definition of a generality order for hypotheses in $\mathcal{L}$ can disregard neither the peculiarities of $\mathcal{SHIQ}$+log$^{\neg}$ nor the methodological apparatus of ILP. One issue arises from the presence of NAF literals (i.e., negated DATALOG literals) both in the background knowledge and in the language of hypotheses. As pointed out in [25], rules in normal logic programs are syntactically regarded as Horn clauses by viewing the NAF-literal $\neg p(X)$ as an atom $not\_p(X)$ with the new predicate not_p. Then any result obtained on Horn logic programs is directly carried over to normal logic programs. Assuming one such treatment of NAF literals, we propose to adapt generalized subsumption [3] to the case of $\mathcal{SHIQ}$+log$^{\neg}$ rules. The resulting generality relation will be called $\mathcal{K}$-*subsumption*, briefly $\succeq_{\mathcal{K}}$, from now on. We provide a characterization of $\succeq_{\mathcal{K}}$ that relies on the reasoning tasks known for $\mathcal{DL}$+log and from which a test procedure can be derived.

Definition 1. *Let $H_1, H_2 \in \mathcal{L}$ be two hypotheses standardized apart, $\mathcal{K}$ a background knowledge, and σ a Skolem substitution for H_2 with respect to $\{H_1\} \cup \mathcal{K}$. We say that $H_1 \succeq_{\mathcal{K}} H_2$ iff there exists a ground substitution θ for H_1 such that (i) $head(H_1)\theta = head(H_2)\sigma$ and (ii) $\mathcal{K} \cup body(H_2)\sigma \models body(H_1)\theta$.*

Note that condition (ii) is a variant of the Boolean CQ/UCQ containment problem because $body(H_2)\sigma$ and $body(H_1)\theta$ are both Boolean CQs. The difference between (ii) and the original formulation of the problem is that $\mathcal{K}$ encompasses not only a TBox but also a set of rules. Nonetheless this variant can be reduced to the satisfiability problem for finite $\mathcal{SHIQ}$+log$^\neg$ KBs. Indeed the skolemization of $body(H_2)$ allows to reduce the Boolean CQ/UCQ containment problem to a CQ answering problem[5]. Due to the aforementioned link between CQ answering and satisfiability, checking (ii) can be reformulated as proving that the KB $(\mathcal{T}, \Pi_R \cup body(H_2)\sigma \cup \{\leftarrow body(H_1)\theta\})$ is unsatisfiable. Once reformulated this way, (ii) can be solved by applying the algorithm NMSAT-$\mathcal{DL}$+log.

Example 6. Let us consider the hypotheses

$H_1^{\texttt{happy}}$	`happy(A)` $\leftarrow$ `RICH(A)`
$H_2^{\texttt{happy}}$	`happy(X)` $\leftarrow$ `famous(X)`

reported in Example 4 up to variable renaming. We want to check whether $H_1^{\texttt{happy}} \succeq_{\mathcal{K}} H_2^{\texttt{happy}}$ holds. Let $\sigma = \{\texttt{X/a}\}$ a Skolem substitution for $H_2^{\texttt{happy}}$ with respect to $\mathcal{K} \cup H_1^{\texttt{happy}}$ and $\theta = \{\texttt{A/a}\}$ a ground substitution for $H_1^{\texttt{happy}}$. The condition (i) is immediately verified. The condition (ii) $\mathcal{K} \cup \{\texttt{famous(a)}\} \models \texttt{RICH(a)}$ is nothing else that a ground query answering problem in $\mathcal{SHIQ}$+log. It can be proved that the query `RICH(a)` can not be satisfied because the rule R1 is not applicable for `a`. Thus, $H_1^{\texttt{happy}} \not\succeq_{\mathcal{K}} H_2^{\texttt{happy}}$. Since $H_2^{\texttt{happy}} \not\succeq_{\mathcal{K}} H_1^{\texttt{happy}}$, the two hypotheses are incomparable under $\mathcal{K}$-subsumption. Conversely, it can be proved that $H_2^{\texttt{happy}} \succeq_{\mathcal{K}} H_3^{\texttt{happy}}$ but not viceversa.

Example 7. Let us consider the hypotheses

$H_1^{\texttt{LONER}}$	`LONER(A)` $\leftarrow$ `scientist(A)`
$H_2^{\texttt{LONER}}$	`LONER(X)` $\leftarrow$ `scientist(X),UNMARRIED(X)`

reported in Example 5 up to variable renaming. We want to check whether $H_1^{\texttt{LONER}} \succeq_{\mathcal{K}} H_2^{\texttt{LONER}}$ holds. Let $\sigma = \{\texttt{X/a}\}$ a Skolem substitution for $H_2^{\texttt{LONER}}$ with respect to $\mathcal{K} \cup H_1^{\texttt{LONER}}$ and $\theta = \{\texttt{A/a}\}$ a ground substitution for $H_1^{\texttt{LONER}}$. The condition (i) is immediately verified. The condition

$$\text{(ii) } \mathcal{K} \cup \{\texttt{scientist(a)}, \texttt{UNMARRIED(a)}\} \models \{\texttt{scientist(a)}\}$$

is a ground query answering problem in $\mathcal{SHIQ}$+log. It can be easily proved that all NM-models for $\mathcal{K} \cup \{\texttt{scientist(a)}, \texttt{UNMARRIED(a)}\}$ satisfy `scientist(a)`. Thus, $H_1^{\texttt{LONER}} \succeq_{\mathcal{K}} H_2^{\texttt{LONER}}$. The viceversa does not hold. Also it can be proved that $H_3^{\texttt{LONER}}$ is incomparable with both $H_1^{\texttt{LONER}}$ and $H_2^{\texttt{LONER}}$ under $\mathcal{K}$-subsumption.

It is straightforward to see that the decidability of $\mathcal{K}$-subsumption follows from the decidability of $\mathcal{SHIQ}$+log$^\neg$. It can be proved that $\succeq_{\mathcal{K}}$ is a quasi-order (i.e. it is a reflexive and transitive relation) for $\mathcal{SHIQ}$+log$^\neg$ rules, therefore the space of hypotheses can be searched by refinement operators.

[5] Since UNA does not necessarily hold in $\mathcal{SHIQ}$, the (Boolean) CQ/UCQ containment problem for $\mathcal{SHIQ}$ boils down to the (Boolean) CQ/UCQ answering problem.

4.2 The hypothesis coverage of observations

The definition of a **coverage relation** depends on the representation choice for observations. An observation $o_i \in O$ is represented as a couple $(p(\boldsymbol{a_i}), \mathcal{F}_i)$ where $\mathcal{F}_i$ is a set containing ground facts concerning the tuple of individuals $\boldsymbol{a_i}$. We assume $\mathcal{K} \cap O = \emptyset$.

Definition 2. *Let $H \in \mathcal{L}$ be a hypothesis, $\mathcal{K}$ a background knowledge and $o_i \in O$ an observation. We say that H* covers o_i under interpretations w.r.t. $\mathcal{K}$ *iff* $\mathcal{K} \cup \mathcal{F}_i \cup H \models p(\boldsymbol{a_i})$.

Note that the coverage test can be reduced to query answering in $\mathcal{SHIQ}$+log$^\neg$ KBs which in its turn can be reformulated as a satisfiability problem of the KB.

Example 8. The hypothesis $H_3^{\texttt{happy}}$ mentioned in Example 4 covers the observation $o_{\texttt{Mary}} = (\texttt{happy(Mary)}, \mathcal{F}_{\texttt{Mary}})$ because $\mathcal{K} \cup \mathcal{F}_{\texttt{Mary}} \cup H_3^{\texttt{happy}} \models \texttt{happy(Mary)}$. Indeed, all NM-models for $\mathcal{B} = \mathcal{K} \cup \mathcal{F}_{\texttt{Mary}} \cup H_3^{\texttt{happy}}$ satisfy:

- `famous(Mary)` (trivial!);
- $\exists$ `WANTS-TO-MARRY`$^-$.$\top$`(Mary)`, due to the axiom A1 and to the fact that both `RICH(Mary)` and `UNMARRIED(Mary)` hold in every model of $\mathcal{B}$;
- `happy(Mary)`, due to the above conclusions and to the rule R1. Indeed, since $\exists$`WANTS-TO-MARRY`$^-$.$\top$`(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.

Note that $H_3^{\texttt{happy}}$ does not cover the observations $o_{\texttt{Joe}} = (\texttt{happy(Joe)}, \mathcal{F}_{\texttt{Joe}})$ and $o_{\texttt{Paul}} = (\texttt{happy(Paul)}, \mathcal{F}_{\texttt{Paul}})$. More precisely, $\mathcal{K} \cup \mathcal{F}_{\texttt{Joe}} \cup H_3^{\texttt{happy}} \not\models \texttt{happy(Joe)}$ because `scientist(Joe)` holds in every model of $\mathcal{B} = \mathcal{K} \cup \mathcal{F}_{\texttt{Joe}} \cup H_3^{\texttt{happy}}$, thus making the rule R1 not applicable for {`X/Joe`}, therefore `RICH(Joe)` not derivable. Finally, $\mathcal{K} \cup \mathcal{F}_{\texttt{Paul}} \cup H_3^{\texttt{happy}} \not\models \texttt{happy(Paul)}$ because `UNMARRIED(Paul)` is not forced to hold in every model of $\mathcal{B} = \mathcal{K} \cup \mathcal{F}_{\texttt{Paul}} \cup H_3^{\texttt{happy}}$, therefore $\exists$`WANTS-TO-MARRY`$^-$.$\top$`(Paul)` is not forced by A1 to hold in every such model.

It can be proved that $H_1^{\texttt{happy}}$ covers $o_{\texttt{Mary}}$ and $o_{\texttt{Paul}}$, while $H_2^{\texttt{happy}}$ all the three observations.

Example 9. With reference to Example 5, the hypothesis $H_3^{\texttt{LONER}}$ does not cover the observation $o_{\texttt{Mary}} = (\texttt{LONER(Mary)}, \mathcal{F}_{\texttt{Mary}})$ because all NM-models for $\mathcal{B} = \mathcal{K} \cup \mathcal{F}_{\texttt{Mary}} \cup H_3^{\texttt{LONER}}$ do satisfy `famous(Mary)`. Note that it does not cover the observations $o_{\texttt{Paul}} = (\texttt{LONER(Paul)}, \mathcal{F}_{\texttt{Paul}})$ and $o_{\texttt{Joe}} = (\texttt{LONER(Joe)}, \mathcal{F}_{\texttt{Joe}})$ for analogous reasons. It can be proved that $H_2^{\texttt{LONER}}$ covers $o_{\texttt{Mary}}$ and $o_{\texttt{Joe}}$ while $H_1^{\texttt{LONER}}$ all three observations.

5 Related Work

Two ILP frameworks have been proposed so far that adopt a hybrid DL-CL representation for both hypotheses and background knowledge. The framework proposed in [24] focuses on discriminant induction and adopts the ILP setting of learning from interpretations. Hypotheses are represented as CARIN-$\mathcal{ALN}$ non-recursive rules with a Horn literal in the head that plays the role of target concept. The coverage relation of hypotheses against examples adapts the usual one in learning from interpretations to the case of hybrid CARIN-$\mathcal{ALN}$ BK. The generality relation between two hypotheses is defined as an extension of generalized subsumption. Procedures for testing both the coverage relation and the generality relation are based on the existential entailment algorithm of CARIN. Following [24], Kietz studies the learnability of CARIN-$\mathcal{ALN}$, thus providing a pre-processing method which enables ILP systems to learn CARIN-$\mathcal{ALN}$ rules [13]. In [15], the representation and reasoning means come from $\mathcal{AL}$-log. Hypotheses are represented as constrained DATALOG clauses. Note that this framework is general, meaning that it is valid whatever the scope of induction is. The generality relation for one such hypothesis language is an adaptation of generalized subsumption to the $\mathcal{AL}$-log KR framework. It gives raise to a quasi-order and can be checked with a decidable procedure based on constrained SLD-resolution. Coverage relations for both ILP settings of learning from interpretations and learning from entailment have been defined on the basis of query answering in $\mathcal{AL}$-log. As opposite to [24], the framework has been partially implemented in an ILP system [16] that supports a variant of frequent pattern discovery where rich prior conceptual knowledge is taken into account in order to find patterns at multiple levels of description granularity.

Table 2. Comparison between ILP frameworks for DL-CL systems.

	Learning in Carin-$\mathcal{ALN}$ [24]	**Learning in $\mathcal{AL}$-log** [15]	**Learning in $\mathcal{SHIQ}$+log$^\neg$**
prior knowledge	CARIN-$\mathcal{ALN}$ KB	$\mathcal{AL}$-log KB	$\mathcal{SHIQ}$+log$^\neg$ KB
ontology lang.	$\mathcal{ALN}$	$\mathcal{ALC}$	$\mathcal{SHIQ}$
rule lang.	Horn clauses	DATALOG clauses	DATALOG$^\neg$ clauses
hypothesis lang.	CARIN-$\mathcal{ALN}$ non-recursive rules	constrained DATALOG clauses	$\mathcal{SHIQ}$+log$^\neg$ non-recursive rules
target predicate	Horn literal	DATALOG literal	$\mathcal{SHIQ}$/DATALOG literal
observations	interpretations	interpretations/implications	interpretations
induction	predictive	predictive/descriptive	predictive/descriptive
generality order	extension of [3] to CARIN-$\mathcal{ALN}$	extension of [3] to $\mathcal{AL}$-log	extension of [3] to $\mathcal{SHIQ}$+log$^\neg$
coverage test	CARIN-$\mathcal{ALN}$ query answering	$\mathcal{AL}$-log query answering	$\mathcal{SHIQ}$+log$^\neg$ query answering
ref. operators	no	downward	no
implementation	no	partially	no
application	no	yes	no

The ILP framework presented in this paper differs from [24] and [15] in several respects as summarized in Table 2, notably the following ones. First, it relies on a more expressive DL, i.e. $\mathcal{SHIQ}$. Second, it allows for inducing definitions for new DL concepts, i.e. rules with a $\mathcal{SHIQ}$ literal in the head. Third, it relies on a more expressive yet decidable CL, i.e. DATALOG$^\neg$. Forth, it adopts a tighter form of integration between the DL and the CL part, i.e. the weakly-safe one. Similarities also emerge from Table 2 such as the use of a *semantic* ordering for hypotheses in order to accommodate ontologies in ILP. Note that generalized subsumption is chosen for adaptation in all three ILP frameworks because definite clauses, though enriched with DL and NAF literals, are still used.

6 Final Remarks

In this paper, we have proposed an ILP framework built upon $\mathcal{SHIQ}$+log$^{\neg}$. Indeed, well-known ILP techniques for induction have been reformulated in terms of the deductive reasoning mechanims of $\mathcal{DL}$+log. Notably, we have defined a decidable generality ordering, $\mathcal{K}$-subsumption, for $\mathcal{SHIQ}$+log$^{\neg}$ rules on the basis of the decidable algorithm NMSAT-$\mathcal{SHIQ}$+log. We would like to point out that the ILP framework proposed is suitable for inductive reasoning in the context of the Semantic Web for two main reasons. First, it adopts the DL which was the starting point for the design of the Web ontology language OWL. Second, it can deal with incomplete knowledge, thus coping with a more plausible scenario of the Web. Though the work presented in this paper can be considered as a feasibility study, it provides the principles for inductive reasoning in $\mathcal{SHIQ}$+log$^{\neg}$. We would like to emphasize that they will be still valid for any other upcoming decidable instantiation of $\mathcal{DL}$+log, provided that DATALOG$^{\neg}$ is still considered for the CL part.

The Semantic Web offers several use cases for rules among which we can choose in order to see our ILP framework at work. As next step towards any practice, we plan to define ILP algorithms starting from the ingredients identified in this paper. Tractable cases, e.g. the instantiation of $\mathcal{DL}$+log with DL-Lite (subset of $\mathcal{SHIQ}$), will be of major interest. Also we would like to investigate the impact of having DATALOG$^{\neg\vee}$ both in the language of hypotheses and in the language for the background theory. The inclusion of the nonmonotonic features of $\mathcal{SHIQ}$+log *full* will strengthen the ability of our ILP framework to deal with incomplete knowledge by performing an inductive form of commonsense reasoning. One such ability can turn out to be useful in the Semantic Web, and complementary to reasoning with uncertainty and under inconsistency. Finally, we would like to study the complexity of $\mathcal{K}$-subsumption.

References

1. 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.
2. A. Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82(1–2):353–367, 1996.
3. W. Buntine. Generalized subsumption and its application to induction and redundancy. *Artificial Intelligence*, 36(2):149–176, 1988.
4. S. Ceri, G. Gottlob, and L. Tanca. What you always wanted to know about datalog (and never dared to ask). *IEEE Transactions on Knowledge and Data Engineering*, 1(1):146–166, 1989.
5. L. De Raedt and L. Dehaspe. Clausal Discovery. *Machine Learning*, 26(2–3):99–146, 1997.

6. F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. $\mathcal{AL}$-log: Integrating Datalog and Description Logics. *Journal of Intelligent Information Systems*, 10(3):227–252, 1998.
7. T. Eiter, G. Gottlob, and H. Mannila. Disjunctive DATALOG. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.
8. A.M. Frisch and A.G. Cohn. Thoughts and afterthoughts on the 1988 workshop on principles of hybrid reasoning. *AI Magazine*, 11(5):84–87, 1991.
9. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.
10. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *Journal of Artificial Intelligence Research*, 31:151–198, 2008.
11. I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
12. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
13. J.-U. Kietz. Learnability of description logic programs. In S. Matwin and C. Sammut, editors, *Inductive Logic Programming*, volume 2583 of *Lecture Notes in Artificial Intelligence*, pages 117–132. Springer, 2003.
14. A.Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104:165–209, 1998.
15. F.A. Lisi. Building Rules on Top of Ontologies for the Semantic Web with Inductive Logic Programming. *Theory and Practice of Logic Programming*, 8(03):271–300, 2008.
16. F.A. Lisi and D. Malerba. Inducing Multi-Level Association Rules from Multiple Relations. *Machine Learning*, 55:175–210, 2004.
17. J.W. Lloyd. *Foundations of Logic Programming*. Springer, 2nd edition, 1987.
18. T.M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.
19. S.-H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Artificial Intelligence*. Springer, 1997.
20. G.D. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.
21. R. Reiter. Equality and domain closure in first order databases. *Journal of ACM*, 27:235–249, 1980.
22. R. Rosati. Semantic and computational advantages of the safe integration of ontologies and rules. In F. Fages and S. Soliman, editors, *Principles and Practice of Semantic Web Reasoning*, volume 3703 of *Lecture Notes in Computer Science*, pages 50–64. Springer, 2005.
23. R. Rosati. $\mathcal{DL}$+log: Tight integration of description logics and disjunctive datalog. In P. Doherty, J. Mylopoulos, and C.A. Welty, editors, *Proc. of Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 68–78. AAAI Press, 2006.
24. C. Rouveirol and V. Ventos. Towards Learning in CARIN-$\mathcal{ALN}$. In J. Cussens and A. Frisch, editors, *Inductive Logic Programming*, volume 1866 of *Lecture Notes in Artificial Intelligence*, pages 191–208. Springer, 2000.
25. C. Sakama. Nonmonotonic inductive logic programming. In T. Eiter, W. Faber, and M. Truszczynski, editors, *Logic Programming and Nonmonotonic Reasoning*, volume 2173 of *Lecture Notes in Computer Science*, pages 62–80. Springer, 2001.
26. M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.