Learning $\mathcal{SHIQ}$+log Rules for Ontology Evolution

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. The definition of new concepts or roles for which extensional knowledge become available can turn out to be necessary to make a DL ontology evolve. In this paper we reformulate this task as a machine learning problem and study a solution based on techniques borrowed from that form of logic-based machine learning known under the name of Inductive Logic Programming (ILP). More precisely, we propose to adapt previous ILP results to the knowledge representation framework of $\mathcal{DL}$+log in order to learn rules to be used for changing $\mathcal{SHIQ}$ ontologies.

1 Introduction

Encoding of human knowledge in ontologies using logical formalisms, e.g. Description Logics (DLs) [1], is one of the crucial tasks to be performed towards the realization of the vision of the Semantic Web. Actually building ontologies is simply the first step because ontologies, just like any structure holding knowledge, need to be maintained as well. Ontology Evolution is the timely adaptation of an ontology to changed business requirements, to trends in ontology instances and patterns of usage of the ontology-based application, as well as the consistent management/propagation of these changes to dependent elements [20]. As opposite to Ontology Modification, Ontology Evolution must preserve the consistency of the ontology. According to [15] one can distinguish between conceptual, specification and representation changes. E.g., modifying a relation is a conceptual change because it affects the conceptualization itself. Also [15] proposes a set of change operations for ontologies considering the effects on the compatibility between two versions of an ontology. E.g., the creation of a class/slot is a lossless change operation because no data is lost. In this paper we consider the conceptual changes of a DL ontology due to extensional knowledge (e.g., facts of the instance level of the ontology) previously unknown but classified which may become available. In particular, we consider the task of defining new concepts or roles which provide the intensional counterpart of this extensional knowledge. One such task, if adequately reformulated, can be partially automated by applying machine learning algorithms. E.g., the new facts `LONER(Joe)`, `LONER(Mary)`, and `LONER(Paul)` concerning known individuals may raise the need for having a definition of the concept `LONER` in the ontology. One such definition can be learned from these facts together with prior knowledge about `Joe`, `Mary` and `Paul`, i.e. facts concerning them and already available in the ontology. A crucial requirement is that the definition must be expressed as a DL formula or similar.

The use of prior or background knowledge (BK) during the learning process is a distinguishing feature of Inductive Logic Programming (ILP) [14]. ILP has been historically concerned with the induction of rules from examples for classification purposes. Unfortunately, it adopts the Knowledge Representation (KR) framework of Logic Programming, i.e. Clausal Logics (CLs) [13], which differ from DLs in several respects. Yet, KR hybrid formalisms exist that combine DLs and CLs. Among the many recent KR proposals, $\mathcal{DL}$+log [16] is a very powerful framework that allows for the tight integration of DLs and disjunctive Datalog with negation (Datalog$^{\neg\vee}$) [6]. A point in favour of $\mathcal{DL}$+log is its decidability for many DLs, notably for $\mathcal{SHIQ}$ [10]. Since the $\mathcal{SH}$ family of very expressive DLs was the starting point for the design of OWL [9], $\mathcal{SHIQ}$+log is a good candidate for investigation in the (Semantic) Web context. 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, denoted as $\mathcal{SHIQ}$+log$^{\neg}$, and adapt ILP techniques to $\mathcal{SHIQ}$+log$^{\neg}$ in order to learn rules that represent the aforementioned conceptual changes for $\mathcal{SHIQ}$ ontologies.

The paper is organized as follows. Section 2 introduces the KR framework of $\mathcal{DL}$+log. Section 3 states the learning problem of interest, defines the core ingredients of an ILP solution to it and sketches an application scenario in Ontology Evolution. Section 4 concludes the paper with final remarks.

2 Representing Rules with $\mathcal{DL}$+log

The KR framework of $\mathcal{DL}$+log [16] allows for the tight integration of DLs [1] and Datalog$^{\neg\vee}$ [6]. 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. In a certain extent, $\mathcal{DL}$+log is between $\mathcal{AL}$-log [5] and Carin [11].

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*.

[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* is a Boolean UCQ with $n = 1$.

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.

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$^{\neg\vee}$ [7]. 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 [16] 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.

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 set G of 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 on NMSAT-$\mathcal{DL}$+log can be found in [16].

[2] This problem was called *existential entailment* in [11].

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 *[16] 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 [8], 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.

3 Learning Concepts and Roles in $\mathcal{SHIQ}$+log$^\neg$ with ILP

3.1 The problem statement

We consider the problem of inducing rule-based definitions of concepts/roles that do not occur in an existing $\mathcal{SHIQ}$ ontology. At this stage of work the scope of induction does not matter. Therefore the term 'observation' is to be preferred to the term 'example'.

Definition 1. *Given:*

- *a $\mathcal{SHIQ}$+log$^\neg$ KB $\mathcal{B}$*
- *a new target $\mathcal{SHIQ}$ predicate name p*
- *a set O of observations for p*
- *a language $\mathcal{L}$ of hypotheses*

the problem of inducing a definition for p is to ***build*** *a hypothesis $\mathcal{H} \in \mathcal{L}$ for p such that $\mathcal{B} \cup \mathcal{H}$ is correct w.r.t. O.*

We assume that the intensional part $\mathcal{K}$ (i.e., the TBox $\mathcal{T}$ plus the set Π_R of rules) of $\mathcal{B}$ plays the role of BK and the extensional part $\mathcal{F}$ (i.e., the ABox $\mathcal{A}$ plus the set Π_F of facts) contributes to the definition of observations. We choose to work within the setting of *learning from interpretations* [4] which requires an observation to be represented as a set of ground unit clauses.

Example 1. Suppose we have a $\mathcal{SHIQ}$+log$^\neg$ KB (adapted from [16]) consisting of the following intensional knowledge $\mathcal{K}$:

$[A1]$ `RICH`$\sqcap$`UNMARRIED` $\sqsubseteq \exists$ `WANTS-TO-MARRY`$^-$.$\top$
$[R1]$ `RICH(X)` $\leftarrow$ `famous(X)`, $\neg$ `scientist(X)`
$[R2]$ `happy(X)` $\leftarrow$ `famous(X)`, `WANTS-TO-MARRY(Y,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 $\mathcal{F}_{\texttt{Paul}} = \{\texttt{famous(Paul)}\}$. Note that $[R2]$ is weakly-safe but not DL-safe because the variable Y does not occur in any Datalog literal of $[R2]$.

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 p is a $\mathcal{SHIQ}$-predicate, each r_j, u_k is a Datalog-predicate, and each s_l is a $\mathcal{SHIQ}$-predicate. Note that p represents the target predicate, i.e. the predicate to be defined by learned $\mathcal{SHIQ}$+log$^\neg$ rules.

Example 2. Suppose that the target predicate is the $\mathcal{SHIQ}$-concept 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) ← UNMARRIED(X)
$H_4^{\texttt{LONER}}$ LONER(X) ← ¬famous(X)

belong to $\mathcal{L}^{\texttt{LONER}}$ and represent hypotheses of definition for LONER.

Example 3. Suppose now that the $\mathcal{SHIQ}$-role LIKES is the target predicate and the set $P_D(\mathcal{L}^{\texttt{LIKES}}) \cup P_C(\mathcal{L}^{\texttt{LIKES}}) \cup P_R(\mathcal{L}^{\texttt{LIKES}}) = \{\texttt{happy/1}\} \cup \{\texttt{RICH/1}\} \cup \{\texttt{WANTS-TO-MARRY/2}\}$ provides the building blocks for the language $\mathcal{L}^{\texttt{LIKES}}$. The following $\mathcal{SHIQ}$+log$^\neg$ rules

$H_1^{\texttt{LIKES}}$ LIKES(X,Y) ← WANTS-TO-MARRY(X,Y)
$H_2^{\texttt{LIKES}}$ LIKES(X,Y) ← WANTS-TO-MARRY(X,Y), happy(X)
$H_3^{\texttt{LIKES}}$ LIKES(X,Y) ← WANTS-TO-MARRY(X,Y), RICH(Y)
$H_4^{\texttt{LIKES}}$ LIKES(X,Y) ← happy(X), RICH(Y)

[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 $\mathit{head}(H)$, 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 $\mathit{head}(H)$ also occur in $\mathit{body}(H)$.

belonging to $\mathcal{L}^{\texttt{LIKES}}$ can be considered hypotheses of definition for LIKES.

Note that a hypothesis $\mathcal{H}$ may consist of more than one $\mathcal{SHIQ}$+log$^{\neg}$ rule. Also $\mathcal{H}$ is valid as a solution to the learning problem in hand if it changes the input ontology by keeping it consistent. This requirement is guaranteed by the correcteness condition in Definition 1.

3.2 The ingredients for an ILP solution

In order to solve the learning problem in hand with the ILP methodological approach , the language $\mathcal{L}$ of hypotheses needs to be equipped with (i) a *generality* order $\succeq$, and (ii) 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.

A generality order for $\mathcal{SHIQ}$+log$^{\neg}$ rules 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 [18], 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 [2] 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 2. *Let $H_1, H_2 \in \mathcal{L}$ be two hypotheses standardized apart, $\mathcal{K}$ a background knowledge, and σ a Skolem substitution*[5] *for H_2 with respect to $\{H_1\} \cup \mathcal{K}$. We say that H_1 is* more general than *H_2 under $\mathcal{K}$-subsumption ($H_1 \succeq_{\mathcal{K}} H_2$) 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. 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.

[5] Let $\mathcal{B}$ be a clausal theory and H be a clause. Let $X_1, \ldots, X_n$ be all the variables appearing in H, and $a_1, \ldots, a_n$ be distinct constants not appearing in $\mathcal{B}$ or H. Then the substitution $\{X_1/a_1, \ldots, X_n/a_n\}$ is called a *Skolem substitution* for H w.r.t. $\mathcal{B}$.

Example 4. Let us consider the hypotheses

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

reported in Example 2 up to variable renaming. We want to check whether $H_1^{\mathtt{LONER}} \succeq_{\mathcal{K}} H_2^{\mathtt{LONER}}$ holds. Let $\sigma = \{\mathtt{X/a}\}$ a Skolem substitution for $H_2^{\mathtt{LONER}}$ with respect to $\mathcal{K} \cup H_1^{\mathtt{LONER}}$ and $\theta = \{\mathtt{A/a}\}$ a ground substitution for $H_1^{\mathtt{LONER}}$. The condition (i) is immediately verified. The condition

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

is a ground query answering problem in $\mathcal{SHIQ}$+log. It can be easily proved that all NM-models for $\mathcal{K} \cup \{\mathtt{scientist(a),UNMARRIED(a)}\}$ satisfy $\mathtt{scientist(a)}$. Thus, $H_1^{\mathtt{LONER}} \succeq_{\mathcal{K}} H_2^{\mathtt{LONER}}$. The viceversa does not hold. Also, $H_3^{\mathtt{LONER}} \succ_{\mathcal{K}} H_2^{\mathtt{LONER}}$ and $H_4^{\mathtt{LONER}}$ is incomparable with all the first three hypotheses.

Example 5. With reference to Example 3, it can be proved that $H_1^{\mathtt{LIKES}} \succ_{\mathcal{K}} H_2^{\mathtt{LIKES}}$ and $H_1^{\mathtt{LIKES}} \succ_{\mathcal{K}} H_3^{\mathtt{LIKES}}$. Conversely, the hypotheses $H_2^{\mathtt{LIKES}}$, $H_3^{\mathtt{LIKES}}$, and $H_4^{\mathtt{LIKES}}$ are incomparable 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.

A coverage relation for $\mathcal{SHIQ}$+log$^\neg$ rules 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 p is the target $\mathcal{SHIQ}$ predicate, $\boldsymbol{a_i}$ is a tuple of individuals occurring in the ABox $\mathcal{A}$ and $\mathcal{F}_i$ is a set containing ground facts concerning individuals in $\boldsymbol{a_i}$. Note that when p is a $\mathcal{SHIQ}$ role name, the tuple $\boldsymbol{a_i}$ is a pair $< a_i^1, a_i^2 >$ of individuals and the set $\mathcal{F}_i$ is given by the union of $\mathcal{F}_i^1$ and $\mathcal{F}_i^2$. We assume $\mathcal{K} \cap O = \emptyset$.

Definition 3. *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 $\mathcal{K} \cup \mathcal{F}_i \cup H \models p(\boldsymbol{a_i})$.*

Therefore 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 6. With reference to Example 2, the hypothesis $H_1^{\mathtt{LONER}}$ covers the observation $o_{\mathtt{Joe}} = (\mathtt{LONER(Joe)}, \mathcal{F}_{\mathtt{Joe}})$ because all NM-models for $\mathcal{B} = \mathcal{K} \cup \mathcal{F}_{\mathtt{Joe}} \cup H_1^{\mathtt{LONER}}$ do satisfy $\mathtt{scientist(Joe)}$. Note that it does not cover the observations $o_{\mathtt{Paul}} = (\mathtt{LONER(Paul)}, \mathcal{F}_{\mathtt{Paul}})$ and $o_{\mathtt{Mary}} = (\mathtt{LONER(Mary)}, \mathcal{F}_{\mathtt{Mary}})$. The hypothesis $H_2^{\mathtt{LONER}}$ convers only $o_{\mathtt{Joe}}$ for analogous reasons. It can be proved that $H_3^{\mathtt{LONER}}$ covers $o_{\mathtt{Mary}}$ and $o_{\mathtt{Joe}}$ while $H_4^{\mathtt{LONER}}$ none of the three observations.

Example 7. None of the hypotheses $H_1^{\mathtt{LIKES}}$, $H_2^{\mathtt{LIKES}}$, and $H_3^{\mathtt{LIKES}}$ reported in Example 3 cover observations concerning couples of known individuals. Conversely, $H_4^{\mathtt{LIKES}}$ covers the observation $o_{\mathtt{<Mary,Paul>}} = (\mathtt{LIKES(Mary,Paul)}, \mathcal{F}_{\mathtt{Mary}} \cup \mathcal{F}_{\mathtt{Paul}})$ because all NM-models for $\mathcal{B} = \mathcal{K} \cup \mathcal{F}_{\mathtt{Mary}} \cup \mathcal{F}_{\mathtt{Paul}} \cup H_4^{\mathtt{LIKES}}$ satisfy:

- `happy(Mary)`, due to the axiom A1 and to the rule R2. Indeed, since from A1 $\exists \mathtt{WANTS\text{-}TO\text{-}MARRY}^{-}.\top\mathtt{(Mary)}$ 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;
- `RICH(Paul)`, since the default rule R1 is always applicable for `Paul`.

Note that $H_4^{\mathtt{LIKES}}$ covers also $o_{\mathtt{<Mary,Mary>}} = (\mathtt{LIKES(Mary,Mary)}, \mathcal{F}_{\mathtt{Mary}})$.

3.3 A proof-of-concept application scenario in Ontology Evolution

The ingredients identified in the previous section are the starting point for the definition of ILP algorithms, that once implemented, can support the evolution of ontologies. Before clarifying how, we remind the reader that - according to [20] - the ontology evolution process is composed of the following six phases:

1. *Change capturing*: This phase encapsulates the process of deciding to apply a change on an ontology. This might be forced by explicit requirements of the ontology engineer or by results of automatic change discovery methods.
2. *Change representation*: In order to resolve changes, they should be identified and represented clearly and in a suitable format.
3. *Semantic of changes*: How a change can affect the ontologys consistency must be understood in advance, whereas the meaning of consistency depends on the underlying ontology model.
4. *Change propagation*: To preserve consistency, affected artefacts should be handled appropriately as well, especially in a distributed environment.
5. *Change implementation*: Before applying a change, all implications of it have to be presented to the user, who then can accept or discard it. If the user agrees with the changes, all activities to apply the change have to be performed.
6. *Change validation*: It should be possible for a user to validate performed changes and to reverse the effects of them when necessary.

We argue that the phases 1.-3. are crucial from our point of view. Indeed change capturing (1.) provides the target predicate and the observations for one or more learning problems of the form as in Definition 1. Once captured this way, each change is represented (2.) as the hypothesis inductively generated according to Definition 1. A particular attention must then be paid to the semantics of those changes (3.) that contain NAF literals because they can affect the ontology consistency. Indeed the change operations considered in this paper, i.e. the creation of a concept and the creation of a role, both boil down to the addition of new rules to the input $\mathcal{SHIQ}$+log$^{\neg}$ KB as illustrated in the following example.

Example 8. Let us suppose that for the concept LONER we have $o_{\texttt{Joe}}$ as a positive example and $o_{\texttt{Mary}}$ and $o_{\texttt{Paul}}$ as negative examples. From this set of observations, an ILP algorithm implementing the ingredients identified in Section 3.2 and adopting a top-down strategy can induce $H_1^{\texttt{LONER}}$ as the hypothesis of rule-based definition for LONER because it covers all positive examples and none of the negative examples w.r.t. the BK of Example 1. Conversely, if $o_{\texttt{Joe}}$ and $o_{\texttt{Mary}}$ are both positive examples and $o_{\texttt{Paul}}$ is the only negative example for LONER, the hypothesis $H_3^{\texttt{LONER}}$ will be returned.

Let us now suppose that $o_{\texttt{<Mary,Paul>}}$ and $o_{\texttt{<Mary,Mary>}}$ are positive examples for the role LIKES and any other observation is considered as negative example. In this case, the hypothesis $H_4^{\texttt{LIKES}}$ is the inductively correct definition of LIKES.

4 Final Remarks

In this paper, we have proposed an ILP framework built upon $\mathcal{SHIQ}$+log$^\neg$ which is a decidable instantiation of the most powerful KR framework currently available for the integration of DLs and CLs. 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 supporting the evolution of a $\mathcal{SHIQ}$ ontology for two main reasons. First, it induces rules with a $\mathcal{SHIQ}$ predicate in the head. Second, it can deal with incomplete knowledge, thus coping with a more plausible scenario of ontology evolution. Though the work presented in this paper can be considered as a feasibility study, it provides the principles for learning 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 ILP framework presented in this paper differs from the related proposals [17] and [12] in several respects, 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$^\neg$). Fourth, it adopts a tighter form of integration between the DL part and the CL part of rules (i.e., the weakly-safe one).

As next step towards any practice, we plan to define ILP algorithms starting from the ingredients identified in this paper. Also, we intend to study in more depth the application of these algorithms to Ontology Evolution, e.g. in the light of related work such as [19] and [3]. Finally, 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 as a complement to reasoning with uncertainty and under inconsistency.

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. W. Buntine. Generalized subsumption and its application to induction and redundancy. *Artificial Intelligence*, 36(2):149–176, 1988.
3. S. Castano, A. Ferrara, and G. Nudelman Hess. Discovery-driven ontology evolution. In G. Tummarello, P. Bouquet, and O. Signore, editors, *Semantic Web Applications and Perspectives*, volume 201 of *CEUR Workshop Proceedings*, 2006.
4. L. De Raedt and S. Džeroski. First order jk-clausal theories are PAC-learnable. *Artificial Intelligence*, 70:375–392, 1994.
5. F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. $\mathcal{AL}$-log: Integrating Datalog and Description Logics. *J. of Intelligent Information Systems*, 10(3):227–252, 1998.
6. T. Eiter, G. Gottlob, and H. Mannila. Disjunctive Datalog. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.
7. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.
8. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *J. of AI Research*, 31:151–198, 2008.
9. 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.
10. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
11. A.Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104:165–209, 1998.
12. 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.
13. J.W. Lloyd. *Foundations of Logic Programming.* Springer, 2nd edition, 1987.
14. S.-H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Artificial Intelligence.* Springer, 1997.
15. N. Fridman Noy and M.C.A. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428–440, 2004.
16. R. Rosati. $\mathcal{DL}$+log: Tight integration of description logics and disjunctive datalog. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proc. of 10th International Conference on Principles of Knowledge Representation and Reasoning*, pages 68–78. AAAI Press, 2006.
17. 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.
18. 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.
19. L. Stojanovic. *Methods and Tools for Ontology Evolution.* Ph.D. Thesis, University of Karlsruhe, 1994.
20. L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-driven ontology evolution management. In A. Gómez-Pérez and V.R. Benjamins, editors, *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, volume 2473 of *Lecture Notes in Computer Science*, pages 285–300. Springer, 2002.