On the Complexity of $\mathcal{EL}$ with Defeasible Inclusions[*]

Piero A. Bonatti
bonatti@na.infn.it

Marco Faella
mfaella@na.infn.it

Luigi Sauro
sauro@na.infn.it

Dep. of Physics, University of Naples "Federico II"

Abstract

We analyze the complexity of reasoning in $\mathcal{EL}$ with defeasible inclusions and extensions thereof. The results by Bonatti et al., 2009a are extended by proving tight lower complexity bounds and by relaxing the syntactic restrictions adopted there. We further extend the old framework by supporting arbitrary priority relations.

1 Introduction

Literature shows a recurrent interest in extending description logics with nonmonotonic reasoning. Early approaches date back to [Brewka, 1987; Straccia, 1993; Baader and Hollunder, 1995a]). Some of the latest motivations originate from the ontologies for biomedical domains and the need of modelling prototypical entities (cf. [Rector, 2004; Stevens et al., 2007]); others arise from the policy languages based on description logics [Uszok et al., 2004; Kagal et al., 2003; Tonti et al., 2003] where nonmonotonic reasoning is needed to express default policies and exceptions [Bonatti et al., 2009a]. So far practical solutions have not been found mainly because of high asymptotic complexity coupled with the absence of effective optimization methods.

Circumscription is particularly interesting in this context. It supports in a natural way specificity-based overriding of inherited properties [Bonatti et al., 2009a], that is, the crucial feature needed to address the aforementioned application requirements.[1] In general, reasoning with circumscribed description logics is very complex, e.g. NExp$^{\mathrm{NP}}$-hard for circumscribed $\mathcal{ALC}$ knowledge bases [Bonatti et al., 2009b]. For this reason, in [Bonatti et al., 2009a] we investigated circumscribed low-complexity description logics of practical interest, such as DL-lite$_R$ [Calvanese et al., 2005] and the $\mathcal{EL}$ family [Baader, 2003; Baader et al., 2005], proving that complexity can be reduced within the second level of the polynomial hierarchy. However, no matching lower complexity bounds were provided.

In this paper we extend our previous results on $\mathcal{EL}^{\perp}$ in several ways. First, the characterization of complexity for the old framework is completed by providing matching hardness results and tightening some upper complexity bounds. Then we extend the framework to support more general queries, general defeasible inclusions (whose left-hand side can be a compound concept), and explicit priorities over defeasible inclusions (that generalize the specificity-based priorities adopted in [Bonatti et al., 2009a]). Finally, we relax the LL restriction adopted in our previous paper (that forbids qualified existentials in the left-hand side of inclusions) by supporting a more liberal use of existential quantification and terminologies (that is, acyclic sets of definitions of the form $A \equiv C$, where C may depend on qualified existential restrictions). This extension covers axioms commonly used by important knowledge bases such as GALEN and SNOMED. Some restrictions on acyclic definitions are still needed to confine complexity within the second level of the polynomial hierarchy. Further contributions will be discussed later.

The rest of the paper is organized as follows: In Section 2, we recall the basics of circumscribed DLs. Sections 3 and 4 deal with the characterization of complexity for variable and fixed concept names, respectively. Section 5 proves that—in some contexts—extensions such as arbitrary partitions in fixed and variable concepts, general defeasible inclusions, and arbitrary priorities do not increase the complexity of reasoning. Section 6 studies the complexity of LL with acyclic terminologies. Two sections on related work and conclusions complete the paper.

2 Syntax and Semantics of Circumscribed $\mathcal{EL}$

We assume the reader to be familiar with the syntax and semantics of monotonic Description Logics. We refer to [Baader et al., 2003, Chap. 2] for details and notation. The sets of *concept names*, *role names*, and *individual names* are denoted by $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$, respectively. By *predicate* we mean any member of $\mathsf{N_C} \cup \mathsf{N_R}$. Hereafter, letters A and B range over $\mathsf{N_C}$, P and R range over $\mathsf{N_R}$, and a, b, c range over $\mathsf{N_I}$. Letters C, D range over concepts.

A *(strong) knowledge base* is a finite set of (i) *concept inclusions (CIs)* $C \sqsubseteq D$, (ii) *concept assertions* $A(a)$, and (iii) *role assertions* $P(a, b)$. Recall that an interpretation $\mathcal{I}$ *satisfies* (i) a CI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, (ii) an assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, (iii) an assertion $R(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. Then, $\mathcal{I}$

[*] This work is very partially supported by the Italian national project LoDeN (http://loden.fisica.unina.it).

[1] This feature is known also as *defeasible inheritance* and is analogous to the inheritance mechanism of object-oriented languages.

is a *model* of a strong knowledge base $\mathcal{S}$ iff $\mathcal{I}$ satisfies all the elements of $\mathcal{S}$. We write $C \sqsubseteq_{\mathcal{S}} D$ iff for all models $\mathcal{I}$ of $\mathcal{S}$, $\mathcal{I}$ satisfies $C \sqsubseteq D$.

The logic $\mathcal{EL}$ [Baader, 2003; Baader et al., 2005] restricts the language to the following constructs:

$$C \quad ::= \quad A \mid \top \mid C_1 \sqcap C_2 \mid \exists P.C$$

The extension of $\mathcal{EL}$ with $\bot$ is denoted by $\mathcal{EL}^{\bot}$.

A *general defeasible inclusion* (GDI) is an expression $C \sqsubseteq_n D$ whose intended meaning is: *C's elements are normally in D.*

Example 2.1 [Bonatti et al., 2009a] The sentences: *"in humans, the heart is usually located on the left-hand side of the body; in humans with situs inversus, the heart is located on the right-hand side of the body"* [Rector, 2004; Stevens et al., 2007] can be formalized with the $\mathcal{EL}^{\bot}$ axioms and GDIs:

$$\begin{array}{l}
\texttt{Human} \sqsubseteq_n \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left}\,; \\
\texttt{Situs\_Inversus} \sqsubseteq \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Right}\,; \\
\exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left} \sqcap \\
\qquad \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Right} \sqsubseteq \bot\,.
\end{array}$$

∎

A *defeasible knowledge base* (DKB) in a logic $\mathcal{DL}$ is a pair $\langle\mathcal{K}, \prec\rangle$, where $\mathcal{K} = \mathcal{K}_S \cup \mathcal{K}_D$, $\mathcal{K}_S$ is a strong $\mathcal{DL}$ KB, $\mathcal{K}_D$ is a set of GDIs $C \sqsubseteq_n D$ such that $C \sqsubseteq D$ is a $\mathcal{DL}$ inclusion, and $\prec$ is a strict partial order (a priority relation) over $\mathcal{K}_D$.

The priority relation $\prec_{\mathcal{K}}$ adopted in [Bonatti et al., 2009a] is based on *specificity*: For all GDIs $\delta_1 = (C_1 \sqsubseteq_n D_1)$ and $\delta_2 = (C_2 \sqsubseteq_n D_2)$, let

$$\delta_1 \prec_{\mathcal{K}} \delta_2 \text{ iff } C_1 \sqsubseteq_{\mathcal{K}_S} C_2 \text{ and } C_2 \not\sqsubseteq_{\mathcal{K}_S} C_1\,.$$

Example 2.2 The access control policy: "*Normally users cannot read project files; staff can read project files; blacklisted staff is not granted any access*" can be encoded with:

$$\begin{array}{l}
\texttt{Staff} \sqsubseteq \texttt{User} \\
\texttt{Blacklisted} \sqsubseteq \texttt{Staff} \\
\texttt{UserRequest} \equiv \exists\texttt{subj}.\texttt{User} \sqcap \exists\texttt{target}.\texttt{Proj} \sqcap \exists\texttt{op}.\texttt{Read} \\
\texttt{StaffRequest} \equiv \exists\texttt{subj}.\texttt{Staff} \sqcap \exists\texttt{target}.\texttt{Proj} \sqcap \exists\texttt{op}.\texttt{Read} \\
\texttt{UserRequest} \sqsubseteq_n \exists\texttt{decision}.\texttt{Deny} \\
\texttt{StaffRequest} \sqsubseteq_n \exists\texttt{decision}.\texttt{Grant} \\
\exists\texttt{subject}.\texttt{Blacklisted} \sqsubseteq \exists\texttt{decision}.\texttt{Deny} \\
\exists\texttt{decision}.\texttt{Grant} \sqcap \exists\texttt{decision}.\texttt{Deny} \sqsubseteq \bot\,.
\end{array}$$

Staff members cannot simultaneously satisfy the two defeasible inclusions (due to the last inclusion above). With specificity, the second defeasible inclusion *overrides* the first one and yields the intuitive inference that non-blacklisted staff members are indeed allowed to access project files. ∎

The semantics of DKBs depends on which predicates are allowed to vary in order to maximize the set of individuals satisfying the GDIs. In [Bonatti et al., 2009a], roles are always allowed to vary, to avoid undecidability problems. The set of concept names $\mathsf{N_C}$, on the contrary, can be arbitrarily partitioned into two sets F and V containing fixed and varying predicates, respectively. Fixed predicates retain their classical semantics, while varying predicates can be affected by nonmonotonic inferences.

The set F, the GDIs $\mathcal{K}_D$, and their ordering $\prec$ induce a strict preference ordering over interpretations, defined below. Roughly speaking, $\mathcal{I}$ is preferred to $\mathcal{J}$ if some GDIs are satisfied by more individuals in $\mathcal{I}$ than in $\mathcal{J}$, possibly at the cost of satisfying less lower-priority GDIs. Formally, for all $\delta = (C \sqsubseteq_n D)$ and all interpretations $\mathcal{I}$ let the set of individuals *satisfying* δ be:

$$\mathsf{sat}_{\mathcal{I}}(\delta) = \{x \in \Delta^{\mathcal{I}} \mid x \notin C^{\mathcal{I}} \text{ or } x \in D^{\mathcal{I}}\}\,.$$

Definition 2.3 Let $\mathcal{KB} = \langle\mathcal{K}, \prec\rangle$ be a DKB. For all interpretations $\mathcal{I}$ and $\mathcal{J}$, and all $F \subseteq \mathsf{N_C}$, let $\mathcal{I} <_{\mathcal{KB},F} \mathcal{J}$ iff:

1. $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$;
2. $a^{\mathcal{I}} = a^{\mathcal{J}}$, for all $a \in \mathsf{N_I}$;
3. $A^{\mathcal{I}} = A^{\mathcal{J}}$, for all $A \in F$;
4. for all $\delta \in \mathcal{K}_D$, if $\mathsf{sat}_{\mathcal{I}}(\delta) \not\supseteq \mathsf{sat}_{\mathcal{J}}(\delta)$ then there exists $\delta' \in \mathcal{K}_D$ such that $\delta' \prec \delta$ and $\mathsf{sat}_{\mathcal{I}}(\delta') \supset \mathsf{sat}_{\mathcal{J}}(\delta')$;
5. there exists a $\delta \in \mathcal{K}_D$ such that $\mathsf{sat}_{\mathcal{I}}(\delta) \supset \mathsf{sat}_{\mathcal{J}}(\delta)$.

The subscript $\mathcal{KB}$ will be omitted when clear from context. Now a *model* of a DKB can be defined as a maximally preferred model of its strong (i.e. classical) part.

Definition 2.4 [Model] Let $\mathcal{KB} = \langle\mathcal{K}, \prec\rangle$ and $F \subseteq \mathsf{N_C}$. An interpretation $\mathcal{I}$ is a *model* of $\mathsf{Circ}_F(\mathcal{KB})$ iff $\mathcal{I}$ is a (classical) model of $\mathcal{K}_S$ and for all models $\mathcal{J}$ of $\mathcal{K}_S$, $\mathcal{J} \not<_F \mathcal{I}$.

In order to enhance readability, $<_{\mathsf{var}}$ and $\mathsf{Circ}_{\mathsf{var}}$ stand for $<_{\emptyset}$ and $\mathsf{Circ}_{\emptyset}$, respectively; $<_{\mathsf{fix}}$ and $\mathsf{Circ}_{\mathsf{fix}}$ stand respectively for $<_{\mathsf{N_C}}$ and $\mathsf{Circ}_{\mathsf{N_C}}$. The standard reasoning tasks are adapted to DKBs as follows:[2]

Concept consistency Given a concept C and a DKB $\mathcal{KB}$, check whether C is *satisfiable w.r.t.* $\mathcal{KB}$, that is, there exists a model $\mathcal{I}$ of $\mathsf{Circ}_F(\mathcal{KB})$ such that $C^{\mathcal{I}} \neq \emptyset$.

Subsumption Given two concepts C, D and a DKB $\mathcal{KB}$, check whether $\mathsf{Circ}_F(\mathcal{KB}) \models C \sqsubseteq D$, that is, for all models $\mathcal{I}$ of $\mathsf{Circ}_F(\mathcal{KB})$, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Instance checking Given $a \in \mathsf{N_I}$, a concept C, and a DKB $\mathcal{KB}$, check whether $\mathsf{Circ}_F(\mathcal{KB}) \models C(a)$, that is, for all models $\mathcal{I}$ of $\mathsf{Circ}_F(\mathcal{KB})$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$.

It has been proved in [Bonatti et al., 2009a] that reasoning in circumscribed $\mathcal{EL}^{\bot}$ is ExpTime-hard, and that complexity decreases for *left local* $\mathcal{EL}^{\bot}$ knowledge bases (LL $\mathcal{EL}^{\bot}$, for short). This fragment consists of the following schemata:

$$\begin{array}{ccc}
A \sqsubseteq_{[n]} B & A \sqsubseteq_{[n]} \exists P.B & A_1 \sqcap A_2 \sqsubseteq B \\
\exists P \sqsubseteq B & \exists P_1 \sqsubseteq \exists P_2.B &
\end{array}$$

where A can be either a concept name or $\top$, and B either a concept name or $\bot$. A *LL* $\mathcal{EL}^{\bot}$ *concept* is any concept that can occur in the above inclusions. It has been proved that if $\mathcal{KB}$ is a LL $\mathcal{EL}^{\bot}$ knowledge base, then LL $\mathcal{EL}^{\bot}$ subsumption and instance checking[3] are in Π^p_2, and LL $\mathcal{EL}^{\bot}$ concept consistency in Σ^p_2.

[2] KB consistency is equivalent to its classical counterpart [Bonatti et al., 2009a], therefore it will not be dealt with in this paper.

[3] Subsumption is LL $\mathcal{EL}^{\bot}$ if C and D are LL $\mathcal{EL}^{\bot}$ concepts; instance checking and concept consistency are LL $\mathcal{EL}^{\bot}$ if C is. Similar terminology applies to other fragments.

3 Complexity of $\mathsf{Circ}_{\mathsf{var}}(LL\ \mathcal{EL}^{\bot})$

Now we prove that reasoning in $\mathsf{Circ}_{\mathsf{var}}(LL\ \mathcal{EL}^{\bot})$ is actually hard (and hence complete) for Σ_2^p and Π_2^p. For this purpose, we provide a reduction of minimal entailment over positive, propositional disjunctive logic programs (PDLP for short), that was proved to be Π_2^p hard in [Eiter and Gottlob, 1995, Theorem 5]. A PDLP over a set of propositional variables $PV = \{p_1, \dots, p_n\}$ is a set of clauses $S = \{c_1, \dots, c_m\}$ over PV where each c_j contains at least one positive literal. The minimal-entailment problem can be then defined as follows: given a literal l, $S \models_{\min} l$ if and only if every $\subseteq$-minimal Herbrand model I of S satisfies l.

For each propositional variable p_i, $1 \leq i \leq n$, introduce two concept names P_i and $\bar{P}_i$ – where the latter encodes $\neg p_i$. In the following we will denote by L_j, $1 \leq j \leq 2n$, a generic P_i or $\bar{P}_i$. For each clause $c_j \in S$ introduce a concept name C_j. Then, two other concept names True and False represent the set of true and false literals respectively. Finally, the concept names Lit and Min are used to model minimal propositional assignments; we need also an auxiliary role R.

First, literals are reified, i.e. modelled as individuals, with the axioms:

$$\top \sqsubseteq (\exists R.L_1 \sqcap \dots \sqcap \exists R.L_{2n}) \qquad (1)$$

$$L_i \sqcap L_j \sqsubseteq \bot \qquad (1 \leq i < j \leq 2n) \qquad (2)$$

$$L_i \sqsubseteq_n \bot \qquad (1 \leq i \leq 2n) \qquad (3)$$

The first axiom makes all L_i nonempty. Axioms (2) make them pairwise disjoint. Finally, axioms (3) minimize the L_i and make them singletons. Then, we represent S by adding for each clause $c_j = l_{j1} \vee \dots \vee l_{jh}$, $1 \leq j \leq m$, the axioms

$$L_{ji} \sqsubseteq C_j \qquad (1 \leq j \leq m \text{ and } 1 \leq i \leq h) \qquad (4)$$

$$C_j \sqsubseteq_n \bot \qquad (1 \leq j \leq m) \qquad (5)$$

$$\top \sqsubseteq \exists R.(C_j \sqcap \mathrm{True}) \qquad (1 \leq j \leq m) \qquad (6)$$

By axioms (4) and (5), C_j equals the set of literals in c_j. Axioms 6 make sure that each clause holds.

In order to model the concepts True and False and the correct meaning of complementary literals we add the axioms

$$\mathrm{True} \sqcap \mathrm{False} \sqsubseteq \bot \qquad (7)$$

$$P_i \sqcap \mathrm{True} \sqsubseteq \exists R.(\bar{P}_i \sqcap \mathrm{False}) \qquad (1 \leq i \leq n) \qquad (8)$$

$$P_i \sqcap \mathrm{False} \sqsubseteq \exists R.(\bar{P}_i \sqcap \mathrm{True}) \qquad (1 \leq i \leq n) \qquad (9)$$

$$\bar{P}_i \sqcap \mathrm{True} \sqsubseteq \exists R.(P_i \sqcap \mathrm{False}) \qquad (1 \leq i \leq n) \qquad (10)$$

$$\bar{P}_i \sqcap \mathrm{False} \sqsubseteq \exists R.(P_i \sqcap \mathrm{True}) \qquad (1 \leq i \leq n) \qquad (11)$$

The axioms defined so far encode the classical semantics of S. To minimize models, add the following axioms:

$$\mathrm{Min} \sqcap P_i \sqsubseteq \mathrm{False} \qquad (1 \leq i \leq n) \qquad (12)$$

$$\mathrm{Min} \sqcap \bar{P}_i \sqsubseteq \mathrm{True} \qquad (1 \leq i \leq n) \qquad (13)$$

$$L_i \sqsubseteq \mathrm{Lit} \qquad (1 \leq i \leq 2n) \qquad (14)$$

$$C_j \sqsubseteq \mathrm{Lit} \qquad (1 \leq j \leq m) \qquad (15)$$

$$\mathrm{Lit} \sqsubseteq_n \mathrm{Min} \qquad (16)$$

By (12) and (13), Min collects false positive literals and true negative literals. By (14) and (15), Lit contains all the (representations of) literals and clauses. The purpose of these axioms is giving defeasible inclusions (3) and (5) higher (specificity-based) priority than (16), so that model minimization cannot cause any L_i to be larger than a singleton, nor any C_j to be different from the set of literals of c_j. Now (16) *prefers* those models where as many P_i as possible are in False.

In the following, given a PDLP S, let $\mathcal{KB}_S$ be the Tbox defined above.

Lemma 3.1 *Given a PDLP S, a literal l in S's language, and the encoding L of l, the following are equivalent:*

(minimal entailment) $S \models_{min} l$

(subs) $\mathsf{Circ}_{\mathsf{var}}(\mathcal{KB}_S) \models \top \sqsubseteq \exists R.(\mathrm{True} \sqcap L)$*;*

(co-sat) $\mathrm{False} \sqcap L$ *is not satisfiable w.r.t* $\mathsf{Circ}_{\mathsf{var}}(\mathcal{KB}_S)$*;*

(instance checking) $\mathsf{Circ}_{\mathsf{var}}(\mathcal{KB}_S) \models (\exists R.(\mathrm{True} \sqcap L))(a)$.

The conjunctions ($\sqcap$) nested in $\exists$ can be easily replaced with a new atom A by adding the equivalence $A \equiv \mathrm{True} \sqcap L$, that can itself be encoded in LL $\mathcal{EL}^{\bot}$, so we have:

Theorem 3.2 *Subsumption and instance checking over* $\mathsf{Circ}_{\mathsf{var}}(LL\ \mathcal{EL}^{\bot})$ *are Π_2^p-hard; concept satisfiability is Σ_2^p-hard. These results hold even if queries are restricted to LL $\mathcal{EL}^{\bot}$ concepts, and priorities are specificity-based.*

4 Complexity of $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{EL}^{\bot})$

When concept names are all fixed, reasoning in LL $\mathcal{EL}^{\bot}$ is tractable:

Theorem 4.1 *$LL\ \mathcal{EL}^{\bot}$ subsumption, instance checking, and concept consistency over* $\mathsf{Circ}_{\mathsf{fix}}(LL\ \mathcal{EL}^{\bot})$ *DKBs are in P.*

The proof is based on a deterministic version of Algorithm 1 (see below), and exploits the fact that defeasible inclusions cannot interfere with each other in $LL\ \mathcal{EL}^{\bot}$, regardless of $\prec$ (details are omitted due to space limitations).

Complexity is low under $\mathsf{Circ}_{\mathsf{fix}}$ because in this context LL axioms are not general enough to simulate quantifier nesting nor conjunctions of existential restrictions. In $\mathsf{Circ}_{\mathsf{var}}$ these features can be simulated by abbreviating compound concepts C with concept names A using equivalences $A \equiv C$ such that C does not depend on qualified existentials (hence the LL restriction is preserved). With $\mathsf{Circ}_{\mathsf{fix}}$, such equivalences change the semantics of C whenever C is (or contains) an existential restriction, because A is fixed and prevents C from varying freely. As we reintroduce the missing features, complexity increases again.

Let $LL_2\mathcal{EL}^{\bot}$ support the schemata:

$$\begin{array}{cc} A \sqsubseteq_{[n]} \exists P.B & \exists P_1 \sqcap \exists P_2 \sqsubseteq \exists P_3.B \\ A \sqsubseteq \exists P.B & \exists P \sqsubseteq B \end{array}$$

The reader may easily verify that $LL_2\mathcal{EL}^{\bot}$ is equivalent to $LL\ \mathcal{EL}^{\bot}$ plus schema $\exists P_1 \sqcap \exists P_2 \sqsubseteq \exists P_3.B$. The missing axioms can be reformulated using fresh roles R and suitable equivalences $\exists R \equiv C$ (that preserve C's semantics because R is a varying predicate).

Theorem 4.2 *Subsumption and instance checking over* $\mathsf{Circ}_{\mathsf{fix}}(LL_2\mathcal{EL}^{\bot})$ *are coNP-hard; concept satisfiability is NP-hard. These results hold even if queries are restricted to $LL_2\mathcal{EL}^{\bot}$ concepts, and priorities are specificity-based.*

Proof. By reduction of SAT. For each propositional variable p_i introduce the concept names A_i, $\bar{A}_i$, and role U_i, representing p_i's truth value (resp. true, false, and undefined). These alternatives are made mutually inconsistent with:

$$A_i \sqcap \bar{A}_i \sqsubseteq \bot \qquad A_i \sqcap \exists U_i \sqsubseteq \bot \qquad \bar{A}_i \sqcap \exists U_i \sqsubseteq \bot$$

For each given clause $c_j = l_{j,1} \vee \cdots \vee l_{j,n}$, introduce a concept name $\bar{C}_j$ representing c_j's falsity. Add $\bar{L}_{j,1} \sqcap \cdots \sqcap \bar{L}_{j,n} \sqsubseteq \bar{C}_j$, where $\bar{L}_{j,k}$ represents the complement of $l_{j,k}$ ($k = 1, \ldots, n$).

Define a concept name $\bar{F}$ representing the falsity of the given set of clauses, and a disjoint concept F with:

$$\bar{C}_j \sqsubseteq \bar{F} \quad \text{(for all input clauses } c_j\text{)} \qquad \bar{F} \sqcap F \sqsubseteq \bot\,.$$

Now, with a defeasible inclusion, $\exists U_i$ is forced to be true for all individuals that satisfy neither A_i nor $\bar{A}_i$; moreover, a role U detects undefined literals:

$$\top \sqsubseteq_n \exists U_i \qquad \exists U_i \sqsubseteq \exists U\,.$$

Let $\mathcal{K}$ be the set of above inclusions and $\mathcal{KB} = \langle \mathcal{K}, \prec_{\mathcal{K}} \rangle$. It can be proved that the given set of clauses S is unsatisfiable iff $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models F \sqsubseteq \exists U$, therefore subsumption checking is coNP-hard.

Similarly, it can be proved that S is unsatisfiable iff $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}') \models (\exists U)(a)$, where $\mathcal{KB}' = \langle \mathcal{K}', \prec_{\mathcal{K}'} \rangle$ and $\mathcal{K}' = \mathcal{K} \cup \{F(a)\}$; therefore instance checking is coNP-hard.

Finally, it can be proved that S is satisfiable iff $F \sqcap \exists OK$ is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}'')$, where $\mathcal{KB}'' = \langle \mathcal{K}'', \prec_{\mathcal{K}''} \rangle$ and $\mathcal{K}'' = \mathcal{K} \cup \{\exists U \sqcap \exists OK \sqsubseteq \bot\}$; therefore satisfiability checking is NP-hard. We are only left to remark that $\mathcal{K}$ can be easily encoded in $LL_2\mathcal{EL}^\perp$. ∎

We prove that this bound is tight using Algorithm 1. There, we assume without loss of generality that $C = A_C \sqcap \bigsqcap_{i=1}^n \exists P_i.B_i$. The algorithm nondeterministically looks for an individual x (in some model) that satisfies C and not D. S_1 guesses any additional fixed concept names satisfied by x; S_2 guesses the concept names that are satisfied somewhere in the model (not necessarily by x) and finally $\prec'$ guesses a total extension of $\prec$ that determines the application order of GDIs.

The algorithm selects, according to $\prec'$, the defeasible inclusions that are *active* in x and accumulates the rhs of those that are not blocked. A defeasible inclusion $A \sqsubseteq_n \exists R.B$ is blocked only if $\exists R.B$ (i) entails locally $\bot$ or a concept name not satisfied by x, or (ii) entails globally the non-emptyness of a concept name that should be empty. The rationale behind blocking is that concept names are fixed and circumscription cannot change their extension as the application of $A \sqsubseteq_n \exists R.B$ would instead require in cases (i) and (ii). Conditions (i)-(ii) are checked, respectively, by means of the sets of concepts $\mathsf{SupCls}(X)$ (the set of atomic concepts that subsume X) and $\mathit{NonEmpty}(X, \mathcal{K}_S)$ (the set of concept names that must be nonempty if X is).

Note that the variable part of C (i.e. $\bigsqcap_{i=1}^n \exists P_i.B_i$) is introduced in X only in line 8, after all defeasible inclusions have been applied, because defeasible inclusions can influence the variable part (e.g. by forcing it to be empty).

Theorem 4.3 *$LL_2\mathcal{EL}^\perp$ subsumption and instance checking over $\mathsf{Circ}_{\mathsf{fix}}(LL_2\mathcal{EL}^\perp)$ are in coNP; $LL_2\mathcal{EL}^\perp$ concept satisfiability is in NP.*

Algorithm 1:

```
  Data: $C = A_C \sqcap \bigsqcap_1^n \exists P_i.B_i,\ D,\ \mathcal{KB} = \langle \mathcal{K}, \prec \rangle$.
1 Guess $S_1, S_2 \subseteq \mathsf{N_C}$, where $\sqcap S_1 \models A_C$ and $S_1 \subseteq S_2$, and a
  linearization $\prec'$ of $\prec$;
2 $\mathsf{A} := \{A \sqsubseteq_n \exists P.B \mid \sqcap S_1 \models A\}$;
3 $\mathsf{X} := \bigsqcap S_1$;
4 while $\mathsf{A} \neq \emptyset$ do
5     remove from $\mathsf{A}$ the $\prec'$-minimal inclusion $A \sqsubseteq_n \exists P.B$;
6     if $\mathsf{SupCls}(\mathsf{X} \sqcap \exists P.B) \subseteq S_1$ and
      $\mathit{NonEmpty}(\mathsf{X} \sqcap \exists P.B, \mathcal{K}_S) \subseteq S_2$ then
7         $\mathsf{X} := \mathsf{X} \sqcap \exists P.B$;
8 $\mathsf{X} := \mathsf{X} \sqcap \bigsqcap_1^n \exists P_i.B_i$;
9 return $\mathsf{SupCls}(\mathsf{X}) \not\subseteq S_1$ or $\mathit{NonEmpty}(\mathsf{X}, \mathcal{K}_S) \not\subseteq S_2$ or
  $\mathsf{X} \sqsubseteq_{\mathcal{K}_S} D$;
```

Proof. (Sketch) Assume without loss of generality that $C = A_C \sqcap \bigsqcap_{i=1}^n \exists P_i.B_i$. It can be proved that Algorithm 1 returns *true* iff the subsumption $C \sqsubseteq D$ is valid. Moreover, $\mathsf{SupCls}(\cdot)$ and $\mathit{NonEmpty}(\cdot,\cdot)$ can be computed in polynomial time using standard $\mathcal{EL}$ reasoning. The theorem for subsumption immediately follows. Concept consistency can be solved by setting $D = \bot$. An instance checking problem $D(a)$ can be solved by collecting in C all the properties of a classically entailed by $\mathcal{K}_S$. ∎

It can be verified that the LL_2 fragment does not support quantifier nesting.

Let *full LL* $\mathcal{EL}^\perp$ (in symbols, $LL_f\mathcal{EL}^\perp$) support all the inclusions $A \sqsubseteq_n D$ and $C \sqsubseteq D$ such that $A \in \mathsf{N_C} \cup \{\top\}$, C and D are $\mathcal{EL}^\perp$ concepts, and no qualified existentials occur in C. The reader may easily verify that full LL $\mathcal{EL}^\perp$ is equivalent to $LL_2\mathcal{EL}^\perp$ plus quantifier nesting. The exact complexity of $\mathsf{Circ}_{\mathsf{fix}}(LL_f\mathcal{EL}^\perp)$ is still unknown. A lower bound is provided in the next section.

5 Complexity of $\mathsf{Circ}_F(\mathcal{EL}^\perp)$ and Extensions

The circuscribed $\mathcal{EL}^\perp$ introduced in Section 2 generalizes the definitions of [Bonatti et al., 2009a] in two ways: First, GDIs can be freely prioritized through $\prec$, and are not restricted to $\prec_K$ (specificity-based priority). Second, in a GDI $C \sqsubseteq_n D$, C is not restricted to concept names. These extensions do not increase complexity. We first show how to encode extended DKBs in the old framework using additional fixed concepts.

Let $\mathcal{KB} = \langle \mathcal{K}, \prec \rangle$ be any given DKB in $\mathcal{EL}^\perp$. First we need to define a new fixed concept A_Δ that encodes the domain without being equivalent to $\top$. This requires the following transformation:

$$\begin{array}{ll}
A^* = A_\Delta \sqcap A & (\exists R.C)^* = A_\Delta \sqcap \exists R.(A_\Delta \sqcap C^*) \\
\top^* = A_\Delta & (C \sqcap D)^* = C^* \sqcap D^* \\
\bot^* = \bot & (C \sqsubseteq_{[n]} D)^* = C^* \sqsubseteq_{[n]} D^* \\
(\exists R)^* = A_\Delta \sqcap \exists R & \text{(only if } \mathcal{KB} \text{ is } LL_f\text{).}
\end{array}$$

Obtain $\mathcal{K}^*$ from $\mathcal{K}$ by transforming all inclusions in $\mathcal{K}$ and by adding a nonemptiness axiom $\top \sqsubseteq \exists aux.A_\Delta$ (aux a fresh role) plus an assertion $A_\Delta(a)$ for each $a \in \mathsf{N_I}$ occurring in $\mathcal{K}$. It is not hard to see that the restrictions to A_Δ of the models of $\mathcal{K}^*$ correspond to the classical models of $\mathcal{K}$. Now we have to remove the new features. For all GDIs $\delta = (C \sqsubseteq_n D) \in$

$\mathcal{K}^*_D$, add two fresh predicates A_δ, R_δ and replace δ with the following axiom schemata:

$$A_\Delta \sqsubseteq A_\delta \qquad A_{\delta'} \sqsubseteq A_\delta \quad \text{for all } \delta' \prec \delta\,. \tag{17}$$

$$A_\delta \sqsubseteq_n \exists R_\delta \qquad \exists R_\delta \sqcap C \sqsubseteq D\,. \tag{18}$$

Call the new DKB $\mathcal{KB}' = \langle \mathcal{K}', \prec_{\mathcal{K}'} \rangle$. By (17), the specificity-based relation $\prec_{\mathcal{K}'}$ prioritizes the new GDIs according to the original priorities. It is not difficult to verify that all the reasoning tasks such that none of the new predicates A_δ and R_δ occur in the query yield the same answer in $\langle \mathcal{K}^*, \prec^* \rangle$ and $\mathcal{KB}'$. Note that the above transformations preserve the $LL_f\mathcal{EL}^\perp$ format. As a consequence of the above discussion, by combining the transformation $\cdot^*$ and (17), (18), we have:

Theorem 5.1 *Let $\mathcal{DL}$ be either $\mathcal{EL}^\perp$ or $LL_f\mathcal{EL}^\perp$. Reasoning in $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{DL})$ with explicit priorities and GDIs can be reduced in polynomial time to reasoning in $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{DL})$ with only specificity-based priority and defeasible inclusions of the form $A \sqsubseteq_n \exists R$.*

One of the applications of this theorem is proving that Circ_F (and hence $\mathsf{Circ}_{\mathsf{var}}$) can be reduced to $\mathsf{Circ}_{\mathsf{fix}}$ in full LL.

Theorem 5.2 *Let $\mathcal{DL}$ be either $\mathcal{EL}^\perp$ or $LL_f\mathcal{EL}^\perp$. Reasoning in $\mathsf{Circ}_F(\mathcal{DL})$ with explicit priorities and GDIs can be reduced in polynomial time to reasoning in $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{DL})$ with only specificity-based priority and defeasible inclusions of the form $A \sqsubseteq_n \exists R$.*

Proof. (Sketch) First remove all variable concepts by introducing a fresh role name R_A for each $A \notin F$, and uniformly replacing A with $\exists R_A$. Then apply the reduction of Lemma 5.1 to remove explicit priorities and general defeasible inclusions (including those introduced by the elimination of variable predicates). ∎

As a corollary of Theorem 5.2 and Theorem 3.2 we have:

Corollary 5.3 *Subsumption and instance checking over $\mathsf{Circ}_{\mathsf{fix}}(LL_f\mathcal{EL}^\perp)$ are Π^p_2-hard; concept satisfiability is Σ^p_2-hard. These results hold even if queries are restricted to LL $\mathcal{EL}^\perp$ concepts, and priorities are specificity-based.*

6 Supporting Acyclic Terminologies

In this section the expressive power of acyclic terminologies[4] is partially recovered.

Definition 6.1 An $\mathcal{EL}^\perp$ knowledge base $\mathcal{KB} = \langle \mathcal{K}, \prec \rangle$ is *almost LL* (*aLL* for short) iff (i) $\mathcal{K} = \mathcal{K}_{LL} \cup \mathcal{K}_a$, (ii) $\mathcal{K}_{LL}$ is in full LL, (iii) $\mathcal{KB}_a$ is a classical acyclic terminology, and (iv) if a concept name A defined in $\mathcal{K}_a$ occurs in the left-hand side of an inclusion in $\mathcal{K}_{LL}$, then A does not depend (in $\mathcal{K}_a$) on any qualified existential restriction.

Example 6.2 This DKB is in aLL. $\mathcal{K}_a$ consists of the first two equivalences:

$$\begin{array}{ll}
\texttt{Yes} \equiv \exists\texttt{decision.Grant} & \texttt{No} \equiv \exists\texttt{decision.Deny} \\
\multicolumn{2}{l}{\texttt{UserRequest} \sqsubseteq \exists\texttt{subj.User} \sqcap \exists\texttt{target.Proj} \sqcap \exists\texttt{op.Read}} \\
\multicolumn{2}{l}{\texttt{StaffRequest} \sqsubseteq \exists\texttt{subj.Staff} \sqcap \exists\texttt{target.Proj} \sqcap \exists\texttt{op.Read}} \\
\texttt{UserRequest} \sqsubseteq_n \texttt{No} & \texttt{StaffRequest} \sqsubseteq_n \texttt{Yes}\,.
\end{array}$$

In general, a concept name A occurring in a terminology $\mathcal{T}$ can be extended with default properties by means of an inclusion $A \sqsubseteq_n C$ in the following cases: A can be a primitive concept (with no definition in $\mathcal{T}$, e.g. `Grant`), or a concept partially defined by a one-way inclusion (e.g. `UserRequest`), or even a concept with a definition $A \equiv D$ in $\mathcal{T}$, provided that A does not depend on any qualified existentials. ∎

The small model property of LL [Bonatti et al., 2009a] can be extended to $\mathsf{Circ}_{\mathsf{var}}(aLL\,\mathcal{EL}^\perp)$ and more general queries. Let $depth(D)$ be the maximum quantifier nesting level in D.

Lemma 6.3 *Let $\mathcal{KB} = \langle \mathcal{K}, \prec_K \rangle$ be an $aLL\,\mathcal{EL}^\perp$ knowledge base (where $\mathcal{K} = \mathcal{K}_a \cup \mathcal{K}_{LL}$) and let C, D be $\mathcal{EL}^\perp$ concepts. For all models $\mathcal{I} \in \mathsf{Circ}_{\mathsf{var}}(\mathcal{KB})$ and for all $x \in C^\mathcal{I} \setminus D^\mathcal{I}$ there exists a model $\mathcal{J} \in \mathsf{Circ}_{\mathsf{var}}(\mathcal{KB})$ such that* (i) $\Delta^\mathcal{J} \subseteq \Delta^\mathcal{I}$, (ii) $x \in C^\mathcal{J} \setminus D^\mathcal{J}$, *and* (iii) $|\Delta^\mathcal{J}|$ *is $O((|\mathcal{KB}| + |C|)^d)$, where $d = depth(D) + 1 + |\mathcal{K}_a|^2$.*

Consequently, as in [Bonatti et al., 2009a], we can prove:

Theorem 6.4 *In $\mathsf{Circ}_{\mathsf{var}}(aLL\ \mathcal{EL}^\perp)$ with specificity-based priorities concept satisfiability is in Σ^p_2. Moreover, deciding $\mathcal{EL}^\perp$ subsumptions $C \sqsubseteq D$ or instance checking problems $D(a)$ with a constant bound on the quantifier depth of D's unfolding (w.r.t. the given DKB) is in Π^p_2.*

Currently, we do not know whether the bound on quantifier nesting depth is necessary to the above result. However, we can prove that the restriction to $\mathsf{Circ}_{\mathsf{var}}$ and $<_\mathcal{K}$ is essential. By means of a reduction of quantified boolean formulae (QBF) satisfiability, it can be proved that:

Theorem 6.5 *Subsumption checking in $\mathsf{Circ}_{\mathsf{fix}}(aLL\ \mathcal{EL}^\perp)$ is PSPACE-hard, even if quantifier mesting depth is bounded by a constant.*

It can be proved that the same holds for $\mathsf{Circ}_{\mathsf{var}}(aLL\ \mathcal{EL}^\perp)$ with general priorities, that can simulate fixed concepts.

7 Related Work

DLs have been extended with nonmonotonic constructs such as default rules [Straccia, 1993; Baader and Hollunder, 1995a; Baader and Hollunder, 1995b], autoepistemic operators [Donini et al., 1997; Donini et al., 2002], and circumscription [Cadoli et al., 1990; Bonatti et al., 2009b; Bonatti et al., 2009a; Bonatti et al., 2010]. The paper [Bonatti et al., 2009a] and this paper have the same goals and adopt essentially the same framework. Here the results of [Bonatti et al., 2009a] are refined and extended as summarized in the next section. The results of [Bonatti et al., 2009b] concern languages more expressive that those considered here; accordingly, complexity is much higher and may reach NEXPTIME$^{\mathrm{NP}}$. A hybrid of $\mathsf{Circ}_{\mathsf{fix}}$ and closed world assumption has been proved to be in PTIME [Bonatti et al., 2010]. A recent approach based on *rational closures* and $\mathcal{ALC}$ can be found in [Casini and Straccia, 2010]; complexity ranges from PSPACE to EXPTIME. A nonmonotonic version of $\mathcal{EL}$ based on a modal *typicality operator* has been introduced in [Giordano et al., 2009b; Giordano et al., 2009a]. Reasoning is NP-hard; the exact complexity is still unknown.

[4] A finite set of *definitions* $A \equiv C$ is a *terminology* if no A occurs in the left-hand side of two different definitions. Given $A \equiv C$, if B occurs in C then A *directly uses* B; *uses* is the transitive closure of *directly uses*. A terminology is *acyclic* if no A uses itself.

$\mathcal{EL}^{\perp}$ DKBs	*subsumption and instance checking**	*concept consistency*
var LL-aLL	Π_2^p-complete	Σ_2^p-complete
fix LL	PTIME †	
fix LL_2	coNP-complete †	NP-complete †
fix LL_f	Π_2^p-hard	Σ_2^p-hard
fix aLL	PSPACE-hard	

$(*)$ *with constant bound on the quantifier depth of the rhs*
$(\dagger)$ *for $LL_2\mathcal{EL}^{\perp}$ queries only ($=LL\,\mathcal{EL}^{\perp}$ queries)*

Figure 1: Main complexity results for $\mathcal{EL}^{\perp}$

8 Conclusions and Further Work

We proved lower complexity bounds for $\mathsf{Circ}_{\mathsf{var}}$ and Circ_F, matching the Π_2^p and Σ_2^p membership results of [Bonatti et al., 2009a], and extended those upper bounds to more general queries (partially supporting quantifier nesting), general defeasible inclusions, and acyclic definitions (as in aLL knowledge bases); these results are summarized in the upper part of Fig. 1. The lower part summarizes the analysis of $\mathsf{Circ}_{\mathsf{fix}}$, that we proved to be less complex than $\mathsf{Circ}_{\mathsf{var}}$, in some cases, because under $\mathsf{Circ}_{\mathsf{fix}}$, LL is not a normal form for full LL (LL_f). The analysis shows the impact of reintroducing conjunctions of existential restrictions (LL_2) and quantifier nesting (LL_f). In Theorem 5.2 we introduced a general method for eliminating variable concepts, GDIs, and general priorities for sufficiently expressive languages. The exact complexity of $\mathsf{Circ}_{\mathsf{fix}}(LL_f\mathcal{EL}^{\perp})$ is still an open question, and so is the question on whether limiting quantifier depth is essential to the upper bounds. Another open question is whether the NP/coNP upper bounds for LL_2 can be extended beyond $LL_2\mathcal{EL}^{\perp}$ queries. Additional features, such as the constructs supported by $\mathcal{EL}^{++}$, will be the subject of further work.

References

[Baader, 2003] Baader, F. (2003). The instance problem and the most specific concept in the description logic EL w.r.t. terminological cycles with descriptive semantics. In *Proc. of KI 2003*, volume 2821 of *LNCS*, pages 64–78. Springer.

[Baader et al., 2005] Baader, F., Brandt, S., and Lutz, C. (2005). Pushing the EL envelope. In *Proc. of IJCAI-05*, pages 364–369.

[Baader and Hollunder, 1995a] Baader, F. and Hollunder, B. (1995a). Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1):149–180.

[Baader and Hollunder, 1995b] Baader, F. and Hollunder, B. (1995b). Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. Autom. Reasoning*, 15(1):41–68.

[Baader et al., 2003] Baader, F., McGuiness, D. L., Nardi, D., and Patel-Schneider, P. (2003). *The Description Logic Handbook: Theory, implementation and applications*. Cambridge Univ. Pr.

[Bonatti et al., 2009a] Bonatti, P. A., Faella, M., and Sauro, L. (2009a). Defeasible inclusions in low-complexity DLs: Preliminary notes. In Boutilier, C., editor, *IJCAI*, pages 696–701.

[Bonatti et al., 2010] Bonatti, P. A., Faella, M., and Sauro, L. (2010). EL with default attributes and overriding. In *ISWC*, volume 6496 of LNCS, pages 64–79. Springer.

[Bonatti et al., 2009b] Bonatti, P. A., Lutz, C., and Wolter, F. (2009b). The complexity of circumscription in dls. *J. Artif. Intell. Res. (JAIR)*, 35:717–773.

[Brewka, 1987] Brewka, G. (1987). The logic of inheritance in frame systems. In *IJCAI*, pages 483–488.

[Cadoli et al., 1990] Cadoli, M., Donini, F., and Schaerf, M. (1990). Closed world reasoning in hybrid systems. In *Proc. of ISMIS'90*, pages 474–481. Elsevier.

[Calvanese et al., 2005] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2005). DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, pages 602–607.

[Casini and Straccia, 2010] Casini, G. and Straccia, U. (2010). Rational closure for defeasible description logics. In *JELIA*, volume 6341 of LNCS, pages 77–90. Springer.

[Donini et al., 1997] Donini, F. M., Nardi, D., and Rosati, R. (1997). Autoepistemic description logics. In *IJCAI (1)*, pages 136–141.

[Donini et al., 2002] Donini, F. M., Nardi, D., and Rosati, R. (2002). Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225.

[Eiter and Gottlob, 1995] Eiter, T. and Gottlob, G. (1995). On the computational cost of disjunctive logic programming: Propositional case. *Ann. Math. Artif. Intell.*, 15(3-4):289–323.

[Giordano et al., 2009a] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2009a). Prototypical reasoning with low complexity description logics: Preliminary results. In *LPNMR*, volume 5753 of LNCS, pages 430–436. Springer.

[Giordano et al., 2009b] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2009b). Reasoning about typicality in ALC and EL. In Grau, B. C., Horrocks, I., Motik, B., and Sattler, U., editors, *Description Logics*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Kagal et al., 2003] Kagal, L., Finin, T., and Joshi, A. (2003). A policy language for a pervasive computing environment. In *4th IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*.

[Rector, 2004] Rector, A. (2004). Defaults, context, and knowledge: Alternatives for OWL-indexed knowledge bases. In *Proc. of the Pacific Symposium on Biocomputing*, pages 226–237.

[Stevens et al., 2007] Stevens, R., Aranguren, M. E., Wolstencroft, K., Sattler, U., Drummond, N., Horridge, M., and Rector, A. L. (2007). Using OWL to model biological knowledge. *International Journal of Man-Machine Studies*, 65(7):583–594.

[Straccia, 1993] Straccia, U. (1993). Default inheritance reasoning in hybrid KL-ONE-style logics. In *IJCAI*, pages 676–681.

[Tonti et al., 2003] Tonti, G., Bradshaw, J. M., Jeffers, R., Montanari, R., Suri, N., and Uszok, A. (2003). Semantic web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. In *ISWC 2003*, volume 2870 of LNCS, pages 419–437. Springer.

[Uszok et al., 2004] Uszok, A., Bradshaw, J. M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., and Aitken, S. (2004). KAoS policy management for semantic web services. *IEEE Intelligent Systems*, 19(4):32–41.