$\mathcal{EL}$ with Default Attributes and Overriding

P. A. Bonatti, M. Faella, and L. Sauro

Università di Napoli Federico II, Napoli, Italy,
{bonatti,mfaella,sauro}@na.infn.it,
WWW home page: http://people.na.infn.it/~bonatti

Abstract. Biomedical ontologies and semantic web policy languages based on description logics (DLs) provide fresh motivations for extending DLs with nonmonotonic inferences—a topic that has attracted a significant amount of attention along the years. Despite this, nonmonotonic inferences are not yet supported by the existing DL engines. One reason is the high computational complexity of the existing decidable fragments of nonmonotonic DLs. In this paper we identify a fragment of circumscribed $\mathcal{EL}^{\bot}$ that supports attribute inheritance with specificity-based overriding (much like an object-oriented language), and such that reasoning about default attributes is in P.

Keywords: Nonmonotonic description logics, Defeasible inheritance.

1 Introduction

The ontologies at the core of the semantic web — as well as ontology languages like RDF and OWL — are based on fragments of first-order logic and inherit strengths and weaknesses of this well-established formalism. Limitations include monotonicity, and the consequent inability to design knowledge bases (KBs) by describing prototypes whose general properties can be later refined with suitable exceptions. This natural approach is commonly used by biologists and calls for an extension of DLs with defeasible inheritance with overriding (a mechanism normally supported by object-oriented languages) [18, 19]. Another motivation for nonmonotonic DLs stems from the recent development of policy languages based on DLs [21, 13, 22, 17]. DLs nicely capture role-based policies and facilitate the integration of semantic web policy enforcement with reasoning about semantic metadata (which is typically necessary in order to check policy conditions). However, in order to formulate standard default policies such as *open* and *closed* policies,[1] and authorization inheritance with exceptions, it is necessary to adopt a nonmonotonic semantics (see the survey [9] for more details).

Given the massive size of semantic web ontologies and RDF bases, it is mandatory that reasoning in nonmonotonic DLs be possible in polynomial time. Unfortunately, in general nonmonotonic DL, reasoning can be highly complex

[1] If no explicit authorization has been specified for a given access request, then an open policy permits the access while a closed policy denies it.

[11, 12, 8]; the best approaches so far belong to the second level of the polynomial hierarchy [10, 7].

In this paper we identify a fragment of circumscribed DLs that extends $\mathcal{EL}$ with default attributes and inheritance with overriding. Informally, the extension allows us to express defeasbile inclusions such as "the instances of C are *normally* in D", for two concepts C and D. Such axioms can be overridden by more specific inclusions, according to a priority mechanism. Our strategy is preserving the classical semantics of $\mathcal{EL}$ as much as possible, in order to facilitate the adaptation of the existing monotonic ontologies. Our framework restricts nonmonotonic inferences to setting the default attributes of "normal" concept instances, without changing the extension of atomic concepts. We define two slightly nonstandard reasoning tasks to query the properties of normal instances. In general, these reasoning tasks are NP-hard. The main cause of intractability is the presence of conflicting defeasible inclusions, i.e., inclusions that give rise to an inconsistency when applied to the same individual. However, if for all pairs of conflicting inclusions δ_1 and δ_2, with non-comparable priority, there exists a disambiguating, higher priority inclusion that blocks at least one of δ_1 and δ_2, then the time complexity of the reasoning tasks becomes polynomial. We show that the identification of such δ_1 and δ_2 can be carried out in polynomial time; then the disambiguation can be left to the ontology engineer or performed automatically by generating a default that blocks both δ_1 and δ_2.

The paper is organized as follows. In Sec. 2, we recall the basics of circumscribed DLs with defeasible inclusions, using the notation adopted in [7]. In Sec. 2.1 we motivate and define a new reasoning task, tailored to inferring the default properties of concepts. Section 3 is devoted to the complexity analysis of this inference problem for the general case and for the restricted class of KBs outlined above. In Sec. 4 the new task and complexity results are extended to instance checking. A section on related work (Sec. 5) and one summarizing our results and discussing interesting future work (Sec. 6) conclude the paper.

2 Preliminaries

In DLs, *concept expressions* are inductively defined using a set of *constructors* (e.g. $\exists$, $\neg$, $\sqcap$), starting with a set $\mathsf{N_C}$ of *concept names*, a set $\mathsf{N_R}$ of *role names*, a set $\mathsf{N_I}$ of *individual names*, and the constants top $\top$ and bottom $\bot$. In what follows, we will deal with expressions

$$C, D ::= A \mid \top \mid \bot \mid C \sqcap D \mid \neg C \mid \exists R.C\,,$$

where A is a concept name and R a role name. In particular, the logic $\mathcal{EL}^{\bot}$ supports all of the above expressions except negation ($\neg C$). Knowledge bases consist in a (finite) set of concept inclusion assertions of the form $C \sqsubseteq D$ (TBox) and a (finite) set of instance assertions of the form $C(a)$, $R(a, b)$ with $a, b \in \mathsf{N_I}$ (ABox).

The semantics of the above concepts is defined in terms of *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The *domain* $\Delta^{\mathcal{I}}$ is a non-empty set of individuals and the *interpretation*

Name	Syntax	Semantics
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists (d,e) \in R^{\mathcal{I}} : e \in C^{\mathcal{I}}\}$
top	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
bottom	$\bot$	$\bot^{\mathcal{I}} = \emptyset$

Fig. 1. Syntax and semantics of some DL constructs.

function $\cdot^{\mathcal{I}}$ maps each concept name $A \in \mathsf{N_C}$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each role name $R \in \mathsf{N_R}$ to a binary relation $R^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and each individual name $a \in \mathsf{N_I}$ to an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation of arbitrary concepts is inductively defined as shown in Figure 1. An interpretation $\mathcal{I}$ is called a *model* of a concept C if $C^{\mathcal{I}} \neq \emptyset$. If $\mathcal{I}$ is a model of C, we also say that C is *satisfied* by $\mathcal{I}$.

An interpretation $\mathcal{I}$ *satisfies* (i) an inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, (ii) an assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and (iii) an assertion $R(a,b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. Then, $\mathcal{I}$ is a *model* of a knowledge base $\mathcal{S}$ iff $\mathcal{I}$ satisfies all the elements of $\mathcal{S}$.

Here we consider *defeasible* $\mathcal{EL}^{\bot}$ knowledge bases $\mathcal{KB} = (\mathcal{S}, \mathcal{D})$ that consist of a (finite) set of classical axioms (inclusions and assertions) $\mathcal{S}$ and a (finite) set $\mathcal{D}$ of defeasible inclusions (DIs for short). Hereafter, with $C \sqsubseteq_{\mathcal{KB}} D$ we mean that D classically subsumes C, that is $\mathcal{S} \models C \sqsubseteq D$. A classical axiom can be either a normal form axiom [1] or an inclusion/disjointness of existential restrictions:

$$A \sqsubseteq B \qquad A_1 \sqsubseteq \exists R.A_2 \qquad A_1 \sqcap A_2 \sqsubseteq B$$

$$\exists P.A \sqsubseteq B \quad \exists R.A_1 \sqsubseteq \exists S.A_2 \quad \exists R.A_1 \sqcap \exists R.A_2 \sqsubseteq \bot$$

where letters of type A can be either a concept name or $\top$, whereas letters B either a concept name or $\bot$. Defeasible axioms take the form $A_1 \sqsubseteq_n \exists R.A_2$ and can be informally be read as *the instances of* A_1 *are normally in* $\exists R.A_2$.

Example 1. A well-known example of prototypical property in a biomedical domain is reported by Rector [18, 19]: *"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"*. A possible formalization in the above language is:

$$\begin{aligned}
&\texttt{Human} \sqsubseteq_n \exists \texttt{has_heart}.\texttt{LHeart}\\
&\texttt{SitusInversus} \sqsubseteq \texttt{Human} \sqcap \exists \texttt{has_heart}.\texttt{RHeart}\\
&\texttt{LHeart} \sqsubseteq \texttt{Heart} \sqcap \exists \texttt{position}.\texttt{Left}\\
&\texttt{RHeart} \sqsubseteq \texttt{Heart} \sqcap \exists \texttt{position}.\texttt{Right}\,.
\end{aligned}$$

In the absence of functional roles, we prevent humans to have both a `LHeart` and a `RHeart` with the disjointness axiom:

$$\exists \texttt{has\_heart}.\texttt{LHeart} \sqcap \exists \texttt{has\_heart}.\texttt{RHeart} \sqsubseteq \bot\,. \qquad \square$$

The nonmonotonic semantics summarized below follows the circumscriptive approach of [7].

Intuitively, a model of a knowledge base $\mathcal{KB}$ is a classical model of $\mathcal{S}$ that maximizes the set of individuals satisfying the defeasible inclusions in $\mathcal{D}$. Formally, for all defeasible inclusions $\delta = (A \sqsubseteq_n C)$ and all interpretations $\mathcal{I}$, the set of individuals *satisfying* δ is:

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

How such sets *can be maximized* depends on what is allowed to vary in an interpretation. Here we assume that only the extension of roles can vary, whereas the domain and the extension of concept names are assumed to be fixed. This semantics is called $\mathsf{Circ}_{\mathsf{fix}}$.

The reason of this choice is rooted in the goal of having a minimal impact on the classical semantics of DLs. If a concept name A is allowed to vary and has exceptional properties, then A may become empty as illustrated in [8]; in most cases, however, it is undesirable to empty a concept only because it has nonstandard properties. It should be possible to extend an existing ontology with default attributes without incurring in such side effects. With $\mathsf{Circ}_{\mathsf{fix}}$, a subsumption $A \sqsubseteq B$ where A and B are atomic concepts is nonmonotonically valid iff it is classically valid. At the same time, it is possible to infer new inclusions like $A \sqsubseteq \exists R.B$ that specify default properties of A. In other words, $\mathsf{Circ}_{\mathsf{fix}}$ supports default attributes without changing the extension of atomic concepts, as desired.

Maximizing defeasible inclusions may lead to conflicts between defeasible inclusions whose right-hand sides are mutually inconsistent. For this reason, it is useful to provide a means to say that a defeasible inclusion δ_1 has higher priority than another defeasible inclusion δ_2. This can be in general provided explicitly by any partial order over $\mathcal{D}$, but here we focus on an implicit way of defining priorities, namely *specificity*, which is based on classically valid inclusions.[2] For all DIs $\delta_1 = (A_1 \sqsubseteq_n C_1)$ and $\delta_2 = (A_2 \sqsubseteq_n C_2)$, we write

$$\delta_1 \prec \delta_2 \text{ iff } A_1 \sqsubseteq_{\mathcal{KB}} A_2 \text{ and } A_2 \not\sqsubseteq_{\mathcal{KB}} A_1.$$

Example 2. Consider the access control policy: "*Normally users cannot read project files; staff can read project files; blacklisted staff is not granted any access*". In circumscribed $\mathcal{EL}^{\bot}$:

$\mathtt{Staff} \sqsubseteq \mathtt{Users}$
$\mathtt{Blacklisted} \sqsubseteq \mathtt{Staff}$
$\mathtt{UserRequest} \equiv \exists \mathtt{subject.Users} \sqcap \exists \mathtt{target.Projects} \sqcap \exists \mathtt{action.Read}$
$\mathtt{StaffRequest} \equiv \exists \mathtt{subject.Staff} \sqcap \exists \mathtt{target.Projects} \sqcap \exists \mathtt{action.Read}$
$\mathtt{UserRequest} \sqsubseteq_n \exists \mathtt{decision.Deny}$
$\mathtt{StaffRequest} \sqsubseteq_n \exists \mathtt{decision.Grant}$
$\exists \mathtt{subject.Blacklisted} \sqsubseteq \exists \mathtt{decision.Deny}$
$\exists \mathtt{decision.Grant} \sqcap \exists \mathtt{decision.Deny} \sqsubseteq \bot\,.$

[2] Since concept names are all fixed and retain their classical semantics, specificity can be equivalently defined using nonmonotonically valid inclusions instead. The result is the same, for all priority relations over defeasible inclusions.

As usual, $C \equiv D$ abbreviates $C \sqsubseteq D$ and $D \sqsubseteq C$. The two equivalences can be reformulated using normal form axioms (see Example 5). Clearly the two defeasible inclusions cannot be simultaneously satisfied for any staff member (due to the last inclusion above). According to 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. □

We are finally ready to formalize the semantics of KBs with defeasible inclusions. The maximization of the sets $\mathsf{sat}_{\mathcal{I}}(\delta)$ is modelled by means of the following preference relation $<_{\mathcal{D}}$ over interpretations. Roughly speaking, $\mathcal{I} <_{\mathcal{D}} \mathcal{J}$ holds iff $\mathcal{I}$ improves $\mathcal{J}$ by extending the set of individuals that satisfy some defeasible inclusions. More precisely, if $\delta_1 \prec \delta_2$ (i.e., δ_1 has higher priority than δ_2), then the set of individuals satisfying δ_1 may be extended at the cost of restricting those that satisfy δ_2.

Definition 1. *For all interpretations $\mathcal{I}$ and $\mathcal{J}$, let $\mathcal{I} <_{\mathcal{D}} \mathcal{J}$ 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* $A \in \mathsf{N_C}$; *(concept name extensions are fixed)*
4. *for all* $\delta \in \mathcal{D}$, *if* $\mathsf{sat}_{\mathcal{I}}(\delta) \not\supseteq \mathsf{sat}_{\mathcal{J}}(\delta)$ *then there exists* $\delta' \in \mathcal{D}$ *such that* $\delta' \prec \delta$ *and* $\mathsf{sat}_{\mathcal{I}}(\delta') \supset \mathsf{sat}_{\mathcal{J}}(\delta')$;
5. *there exists a* $\delta \in \mathcal{D}$ *such that* $\mathsf{sat}_{\mathcal{I}}(\delta) \supset \mathsf{sat}_{\mathcal{J}}(\delta)$.

The subscript $\mathcal{D}$ will be omitted when clear from the context. Now $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB})$ iff $\mathcal{I}$ is a model of $\mathcal{S}$ that cannot be further improved (defeasible inclusions are satisfied "as much as possible").

Definition 2 (Model). *Let $\mathcal{KB} = (\mathcal{S}, \mathcal{D})$, an interpretation $\mathcal{I}$ is a* model *of $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB})$ iff $\mathcal{I}$ is a (classical) model of $\mathcal{S}$ and for all models $\mathcal{J}$ of $\mathcal{S}$, $\mathcal{J} \not< \mathcal{I}$.*

Example 3. Let $\mathcal{KB}$ be the knowledge base of Example 2. According to condition 2 in Def. 1, model improvements cannot change the extension of atomic concepts;[3] therefore, if `Grant` and `Deny` are empty in a model, then the two defeasible inclusions of $\mathcal{KB}$ cannot possibly force any request to satisfy $\exists$`decision`.`Grant` nor $\exists$`decision`.`Deny`. In order to "enable" the two DIs, it suffices to assert that `Grant` and `Deny` are non-empty, by means of an auxiliary role `aux` and two simple inclusions:

$$\top \sqsubseteq \exists \texttt{aux}.\texttt{Grant} \qquad \top \sqsubseteq \exists \texttt{aux}.\texttt{Deny}\,.^{4}$$

Now the two DIs can "fire" and, as a consequence, the models of $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB})$ are all the models of the classical inclusions of $\mathcal{KB}$ such that for all individuals x satisfying $\exists$`target`.`Projects` $\sqcap$ $\exists$`action`.`Read`,

[3] Recall that this is one of our requirements, aimed at controlling the side effects of adding defeasible inclusions to existing classical ontologies.

[4] These axioms are usually harmless and can be inserted with the help of automated tools, that identify which concepts occurring in the right hand side of a DI can possibly be empty.

– if x satisfies $\exists$`subject.Blacklisted`, then x satisfies $\exists$`decision.Deny`;
– otherwise, if x satisfies $\exists$`subject.Staff`, then x satisfies $\exists$`decision.Grant`;
– otherwise, if x satisfies $\exists$`subject.User`, then x satisfies $\exists$`decision.Deny`. □

The above example shows the need for declaring the non-emptyness of default attribute ranges, such as B in $A \sqsubseteq_n \exists R.B$. In theory, such declarations may be inconsistent with the knowledge base; however, in practice, concept names are usually meant to be non-empty and, accordingly, concept consistency checking is a typical step in ontology validation. In other words, we only need to make explicit some assumptions that are sometimes left implicit; this can be done automatically for all default attribute ranges B. These additional axioms can be easily checked for consistency: In $\mathcal{EL}^\bot$, if all non-emptyness statements are individually consistent with the KB, then also the set of all non-emptyness statements is collectively consistent; consequently, no combinatorial problems arise and consistency checking remains polynomial. It is not difficult to extend this framework with nominals and concrete datatypes; when default attributes range over nominals or concrete domains, non-emptyness is implicit in the logic and no explicit declarations are needed.

2.1 A New Reasoning Task

Now that we have provided constructs for associating concepts to default properties, we need a suitable reasoning task to retrieve them. For example, from the formalization of human heart we would like to infer that typical humans satisfy $\exists$`has_heart.LHeart`. Subsumption queries, according to [8], are defined as follows: $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models C \sqsubseteq D$ iff for all models $\mathcal{I}$ of $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB})$, $C^\mathcal{I} \subseteq D^\mathcal{I}$. This reasoning method is not completely appropriate for our purposes, because a standard subsumption query $A \sqsubseteq \exists R.C$ considers not only the typical members of A, but also the typical members of A's subconcepts, where A's default properties may be overridden. In this way, some of A's default properties might not be included in the answer. For instance, in the context of the *situs inversus* example, it is generally not possible to entail `Humans` $\sqsubseteq \exists$`has_heart.LHeart`, because the members of `Humans` comprise all the members of `SitusInversus`, too, that are forced to satisfy $\exists$`has_heart.RHeart`, instead. For this reason, in this work we consider a slightly modified subsumption problem, according to which a query $A \sqsubseteq \exists R.C$ is interpreted as: "Do the individuals belonging to A *and no subconcepts of* A satisfy $\exists R.C$?". This is a sort of closed world assumption. It is equivalent to interpreting $A \sqsubseteq \exists R.C$ as $CWA_{\mathcal{KB}}(A) \sqsubseteq \exists R.C$, where $CWA_{\mathcal{KB}}(A) = A \sqcap \bigsqcap\{\neg B \mid B \in \mathsf{N_C} \text{ and } A \not\sqsubseteq_{\mathcal{KB}} B\}$. In $\mathcal{EL}^\bot$, this closure cannot introduce any inconsistency:

Theorem 1. *For all $\mathcal{EL}^\bot$ knowledge bases $\mathcal{KB}$, $CWA_{\mathcal{KB}}(A)$ is satisfiable w.r.t. $\mathcal{KB}$ iff A is satisfiable w.r.t. $\mathcal{KB}$.*

$CWA_{\mathcal{KB}}(A)$ can be equivalently defined in purely model theoretic terms not involving $\neg$ as the set $\lfloor A \rfloor^\mathcal{I}$ that denotes the set of all individuals $d \in A^\mathcal{I}$ such that, for all concept names B, $d \in B^\mathcal{I}$ holds only if $A \sqsubseteq_{\mathcal{KB}} B$. Then, we define the modified entailment problem as follows:

Definition 3. *Let* $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} A \sqsubseteq D$ *hold if and only if for all models* $\mathcal{I}$ *of* $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB})$, $\lfloor A \rfloor^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Example 4. Extend the knowledge base of Example 1 with $\top \sqsubseteq \exists \mathtt{aux}.\mathtt{LHeart}$, to ensure that there exists at least one normal heart.[5] Note that

$$CWA_{\mathcal{KB}}(\mathtt{Human}) = \mathtt{Human} \sqcap \neg \mathtt{SitusInversus} \sqcap \neg \mathtt{LHeart} \sqcap \neg \mathtt{RHeart} \sqcap \neg \mathtt{Heart} \sqcap \neg \mathtt{Left} \sqcap \neg \mathtt{Right}.$$

It is not hard to see that $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} \mathtt{Human} \sqsubseteq \exists \mathtt{has\_heart}.\mathtt{LHeart}$ and $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} \mathtt{SitusInversus} \sqsubseteq \exists \mathtt{has\_heart}.\mathtt{RHeart}$, as desired. □

The reader may wonder whether in general the CWA can be too restrictive and miss valid default properties. This might happen if a concept A's extension could be completely covered by n subconcepts $A_1, \ldots, A_n$ sharing a same default property $\exists R.B$. In this case, it would be natural to require A's prototypical members to satisfy $\exists R.B$, as they must necessarily fall into some A_i. However, in $\mathcal{EL}^{\bot}$ such coverings cannot be defined, i.e. there is always a model $\mathcal{I}$ in which there exists $d \in A^{\mathcal{I}} \setminus \bigcup_{i=1}^{n} A_i^{\mathcal{I}}$. Such d need not satisfy $\exists R.B$, and hence it would be inappropriate to list $\exists R.B$ among the default properties of A.

3 Complexity

3.1 NP-hardness of the General Case

In general, deciding whether $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} A \sqsubseteq D$ holds is NP-hard. This can be proved by reducing SAT to our reasoning task. For each clause c_i in the SAT instance introduce two roles C_i and $\bar{C}_i$. Intuitively, the meaning of $\exists C_i$ and $\exists \bar{C}_i$ is: c_i is/is not satisfied, respectively. For each propositional symbol p_j introduce two roles P_j and $\bar{P}_j$. Intuitively, $\exists P_j$ and $\exists \bar{P}_j$ represent the truth of the complementary literals p_j and $\neg p_j$, respectively. Then, we need two concept names B_0 and B_1, and a role $\bar{F}$. Intuitively, $\exists \bar{F}$ represents the falsity of the set of clauses. The semantics of clauses is axiomatized by adding the inclusions

$$\exists P_j \sqsubseteq \exists C_i\,, \quad \exists \bar{P}_k \sqsubseteq \exists C_i\,,$$

for all disjuncts p_j and $\neg p_k$ in c_i. The space of possible truth assignments is generated by the following inclusions:

$$B_0 \sqsubseteq_n \exists P_j\,, \quad B_0 \sqsubseteq_n \exists \bar{P}_j\,, \quad \exists P_j \sqcap \exists \bar{P}_j \sqsubseteq \bot\,.$$

All of the above defaults have the same priority. The defeasible inclusions with the same index j "block" each other; we make at least one of them active by assuming B_0; this "forces" a complete truth assignment. Then we introduce a defeasible inclusion with lower priority:

$$B_0 \sqsubseteq B_1\,, \quad B_1 \sqsubseteq_n \exists \bar{C}_i\,.$$

[5] See Example 3 for an explanation of this kind of axioms.

This defeasible inclusion "assumes" that c_i is not satisfied. The first three groups of axioms may defeat this assumption (if the selected truth assignment entails $\exists C_i$) thanks to the following disjointness axiom:

$$\exists C_i \sqcap \exists \bar{C}_i \sqsubseteq \bot .$$

Finally, we need the inclusions $\exists \bar{C}_i \sqsubseteq \bar{F}$ to say that the set of clauses is not satisfied when at least one of the clauses is false. Now let $\mathcal{KB}$ denote the above set of inclusions. It can be proved that the given set of clauses is unsatisfiable iff:

$$\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} B_0 \sqsubseteq \exists \bar{F} .$$

Consequently:

Theorem 2. *Let $\mathcal{KB}$ range over $\mathcal{EL}^{\bot}$ knowledge bases. The problem of checking whether $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C \sqsubseteq D$ is NP-hard, even if C is a concept name and D an unqualified existential restriction.*

3.2 A Polynomial Case

The above reduction of SAT is based on concepts with equally specific, conflicting default properties. In our reference scenarios, we expect such situations to be symptoms of representation errors. For instance, in modelling prototypical entities, equally specific and conflicting default properties constitute a contradictory prototype definition. In the access control domain, a class of requests associated to conflicting decisions with the same priority constitutes an ambiguous policy, with potentially dangerous consequences. In this section, we focus on a class of KBs called *conflict safe*, where this kind of conflicts cannot occur. This restriction turns out to reduce the computational complexity of reasoning.

Intuitively, the idea is that it is possible to check efficiently whether two defaults δ_1 and δ_2 block each other and none of them is more specific than the other (as in the reduction from SAT). Such conflicts, that make the search space grow, can be solved (either manually or automatically) by adding more specific defaults that determine how to resolve the conflict (either in favor of one of the δ_is or blocking them both). In the following, let $\mathcal{KB} = (\mathcal{S}, \mathcal{D})$ be an arbitrary knowledge base. The next definitions are all relative to $\mathcal{KB}$.

We say that two defeasible inclusions are in conflict when they can be simultaneously activated (their premises are mutually consistent) and their conclusions are mutually inconsistent. The formal definition follows.

Definition 4. *Two defeasible inclusions $\delta_1 = A_1 \sqsubseteq_n \exists R.A_1'$ and $\delta_2 = A_2 \sqsubseteq_n \exists S.A_2'$ are* in conflict*, denoted by $\delta_1 \nleftrightarrow \delta_2$, iff $A_1 \sqcap A_2 \not\sqsubseteq_{\mathcal{KB}} \bot$ and $\exists R.A_1' \sqcap \exists S.A_2' \sqsubseteq_{\mathcal{KB}} \bot$.*

Since classical subsumption in $\mathcal{EL}^{\bot}$ knowledge bases can be computed in polynomial time [2], we have:

Proposition 1. *Given an $\mathcal{EL}^{\bot}$ knowledge base $\mathcal{KB} = (\mathcal{S}, \mathcal{D})$ and two defaults δ_1 and δ_2 in $\mathcal{D}$, the problem of checking whether $\delta_1 \nleftrightarrow \delta_2$ is in* PTIME.

A naive approach to listing all the conflicting pairs, consists in performing a quadratic number of $\mathcal{EL}^{\bot}$ subsumptions. Better strategies can be obtained by adapting the ideas behind efficient classification algorithms [6, Chap. 9] to reduce the number of comparisons (the details lie beyond the scope of this paper). In this section we assume that $\mathcal{KB}$ is *conflict safe* in the following sense:

Definition 5. *$\mathcal{KB}$ is* conflict safe *iff whenever two defeasible inclusions $\delta_1 = A_1 \sqsubseteq_n \exists R.A_1'$ and $\delta_2 = A_2 \sqsubseteq_n \exists S.A_2'$ are incomparable and in conflict (i.e. $\delta_1 \not\prec \delta_2$, $\delta_2 \not\prec \delta_1$ and $\delta_1 \nleftrightarrow \delta_2$), then (i) $A_1 \not\equiv_{\mathcal{KB}} A_2$, (ii) there exists a concept name A_3 such that $A_3 \equiv_{\mathcal{KB}} A_1 \sqcap A_2$, and (iii) one of the following sets of inclusions belongs to $\mathcal{KB}$:*

- $A_3 \sqsubseteq_n \exists R.A_1'$;
- $A_3 \sqsubseteq_n \exists S.A_2'$;
- $A_3 \sqsubseteq_n \exists T$ *and* $\exists T \sqcap \exists R.A_1' \sqsubseteq \bot$ *and* $\exists T \sqcap \exists S.A_2' \sqsubseteq \bot$. □

Note that the above three DIs (whose priority is higher than δ_1 and δ_2) correspond to three possible ways of resolving the conflict between δ_1 and δ_2, namely, supporting the conclusion of δ_1, supporting the conclusion of δ_2, or blocking both δ_1 and δ_2 . The third option constitutes a possible default conflict resolution strategy that can be performed automatically by introducing fresh roles T and the corresponding disjointness axioms. Note also that our two running examples are conflict safe because all conflicting defaults are comparable and specificity resolves the conflict.

We proceed towards a PTIME algorithm for reasoning with conflict safe KBs. We first need some preliminary definitions. Given a concept C, $\mathsf{SupCls}(C)$ denotes the set of *superclasses* of C:

$$\mathsf{SupCls}(C) = \{B \mid C \sqsubseteq_{\mathcal{KB}} B\} \cup \{\exists R.A \mid C \sqsubseteq_{\mathcal{KB}} \exists R.A\}. \tag{1}$$

We write $C \rightsquigarrow A$ if $C \sqsubseteq_{\mathcal{KB}} \exists R.A$ for some R, and we denote by $\overset{*}{\rightsquigarrow}$ the transitive closure of $\rightsquigarrow$. Given a concept C, the operator $\mathsf{NE}(C)$ represents the set of concepts that are forced to be *non-empty* whenever C is. Notice that this set includes some concepts that are forced to be non-empty by the ABox in $\mathcal{KB}$, independently of C.

$$\mathsf{NG}(C) = \{C\} \cup \bigcup_{a \in \mathsf{N_I}} \{A \mid \mathcal{KB} \models A(a)\} \cup \bigcup_{a \in \mathsf{N_I}, R \in \mathsf{N_R}} \{A \mid \mathcal{KB} \models (\exists R.A)(a)\} \tag{2}$$

$$\mathsf{NE}(C) = \bigcup_{A \in \mathsf{NG}(C)} \{A' \mid A \overset{*}{\rightsquigarrow} A'\}. \tag{3}$$

When trying to satisfy a certain defeasible inclusion $A_1 \sqsubseteq_n \exists R.A_2$, we have to check two forms of consistency. First, the addition of an R edge to A_2 should be possible without modifying the interpretation of the concepts names, that are fixed. This check is realized by the following function $\mathsf{Comp}_{\mathrm{fix}}$. Second, the addition of $\exists R.A_2$ should not lead to classical inconsistencies, also considering other defeasible inclusions that were previously satisfied. This check is realized by the function Cons.

Algorithm 1:

Data: C, $\mathcal{KB} = \langle \mathcal{S}, \mathcal{D} \rangle$.

1 $\mathsf{X} \leftarrow \mathsf{SupCls}(C)$;
2 **while** $\mathcal{D} \neq \emptyset$ **do**
3 remove from $\mathcal{D}$ an inclusion $\delta = (A_1 \sqsubseteq_n \exists R.A_2)$ with maximal priority;
4 **if** $A_1 \in \mathsf{SupCls}(C)$ *and* $\delta \in \mathsf{Comp}_{\mathrm{fix}}(C) \cap \mathsf{Cons}(\mathsf{X})$ **then**
5 $\mathsf{X} \leftarrow \mathsf{X} \cup \mathsf{SupCls}(\exists R.A_2)$;
6 **return** X;

For a concept C, $\mathsf{Comp}_{\mathrm{fix}}(C)$ (for *fixed-atoms compatible*) is the set of defeasible inclusions whose r.h.s. agree with C on the inferred and non-empty concept names. That is, a defeasible inclusion $A_1 \sqsubseteq_n \exists R.A_2$ is in $\mathsf{Comp}_{\mathrm{fix}}(C)$ if and only if: *(i)* $\mathsf{NE}(\exists R.A_2) \subseteq \mathsf{NE}(C)$ and *(ii)* for all concept names $A \in \mathsf{SupCls}(\exists R.A_2)$, it holds $A \in \mathsf{SupCls}(C)$.

For a set of concepts X, $\mathsf{Cons}(\mathsf{X})$ is the set of defeasible inclusions whose r.h.s. is logically *consistent* with X. That is, a defeasible inclusion $A_1 \sqsubseteq_n \exists R.A_2$ is in $\mathsf{Cons}(\mathsf{X})$ if and only if $\bigsqcap_{D \in \mathsf{X}} D \sqcap (\exists R.A_2) \not\sqsubseteq_{\mathcal{KB}} \bot$.

We claim that Algorithm 1, when invoked over the concept C, returns the set of all concepts C' that are implied by C under the closed world assumption.

Theorem 3. *Let* X *be the result of Algorithm 1 on the concept* C*. If* $\mathcal{KB}$ *is conflict safe and assertion-free*[6] *then* $\mathsf{X} = \{C' \mid \mathsf{Circ}_{\mathrm{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C \sqsubseteq C'\}$.

Proof. *(⊆)* Let $C' \in \mathsf{X}$. If C' was inserted in line 1 of the algorithm, then C' is classically implied by C (i.e., $C \sqsubseteq_{\mathcal{KB}} C'$), and hence $\mathsf{Circ}_{\mathrm{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C \sqsubseteq C'$. Otherwise, C' was inserted in line 5. Hence, there is a defeasible inclusion $\delta = (A_1 \sqsubseteq_n \exists R.A_2)$ such that $A_1 \in \mathsf{SupCls}(C)$, $\delta \in \mathsf{Comp}_{\mathrm{fix}}(C) \cap \mathsf{Cons}(\mathsf{X}')$ and $C' \in \mathsf{SupCls}(\exists R.A_2)$, where X' is the value of the variable X when C' was inserted. By applying the definition of $\mathsf{Comp}_{\mathrm{fix}}$, we obtain that *(i)* $\mathsf{NE}(\exists R.A_2) \subseteq \mathsf{NE}(C)$, and *(ii)* for all concept names $A' \in \mathsf{SupCls}(\exists R.A_2)$, it holds $A' \in \mathsf{SupCls}(C)$.

Let $\mathcal{I}$ be a model of $\mathsf{Circ}_{\mathrm{fix}}(\mathcal{KB})$ with an individual $d \in \lfloor C \rfloor^{\mathcal{I}}$, we show that $d \in C'^{\mathcal{I}}$. Assume by contradiction that $d \notin C'^{\mathcal{I}}$. Since A_1 is a classical consequence of C, we have $d \in A_1^{\mathcal{I}}$. Since $C' \in \mathsf{SupCls}(\exists R.A_2)$, we have $d \notin (\exists R.A_2)^{\mathcal{I}}$. We show that there exists a classical model $\mathcal{J}$ of $\mathcal{KB}$ that improves $\mathcal{I}$, i.e., $\mathcal{J} <_{\mathcal{D}} \mathcal{I}$. To define $\mathcal{J}$, for all $\exists S.A_3 \in \mathsf{SupCls}(\exists R.A_2)$ (including $\exists R.A_2$ itself), we add to $\mathcal{I}$ an S-arc from d to an individual $x \in A_3^{\mathcal{J}}$. The existence of such an individual is guaranteed by the fact that $\mathsf{NE}(\exists R.A_2) \subseteq \mathsf{NE}(C)$. As a result, we have in particular that $d \in (\exists R.A_2)^{\mathcal{J}}$. By *(ii)*, all atomic concepts that are classical consequences of $\exists R.A_2$ are also consequences of C. This, together with the fact that $\delta \in \mathsf{Cons}(\mathsf{X}')$, ensures that $\mathcal{J}$ is a classical model of $\mathcal{KB}$. It remains to prove that $\mathcal{J} <_{\mathcal{D}} \mathcal{I}$. Since $\mathcal{I}$ and $\mathcal{J}$ only differ on the arcs outgoing from d, we have

[6] In DL jargon: the ABox is empty. The reason for considering ABox assertions in the definition of $\mathsf{NG}(C)$ will be clear in the next section, when we deal with instance checking.

$\mathsf{sat}_{\mathcal{I}}(\delta) \subset \mathsf{sat}_{\mathcal{J}}(\delta)$ and for all $\delta' \neq \delta$ in $\mathcal{D}$, we have $\mathsf{sat}_{\mathcal{I}}(\delta') = \mathsf{sat}_{\mathcal{J}}(\delta')$. Therefore, we obtain the thesis.

($\supseteq$) Let C' be a concept such that $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C \sqsubseteq C'$. Assume by contradiction that C' does not belong to the output X of the algorithm. Clearly, $C \not\sqsubseteq_{\mathcal{KB}} C'$, otherwise C' would have been added to X in step 1 of the algorithm. Since $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C \sqsubseteq C'$, there is a defeasible inclusion $A_1 \sqsubseteq_n \exists R.A_2 \in \mathcal{D}$ such that $A_1 \in \mathsf{SupCls}(C)$ and $\exists R.A_2 \sqsubseteq_{\mathcal{KB}} C'$ Let $\hat{\delta} \in \mathcal{D}$ be a defeasible inclusion with the above property and maximal priority. At some point, $\hat{\delta}$ is extracted from $\mathcal{D}$ at step 3 of the algorithm. Since C' is never added to X, we have that either $\hat{\delta} \notin \mathsf{Comp}_{\mathsf{fix}}(C)$ or $\hat{\delta} \notin \mathsf{Cons}(\mathsf{X}')$, where X' is the current value of the variable X. In both cases, it is possible to define a model $\mathcal{I}$ of $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB})$ with an individual $d \in \Delta^{\mathcal{I}}$ such that $d \in \lfloor C \rfloor^{\mathcal{I}} \setminus C'^{\mathcal{I}}$, which is a contradiction.

We define $\mathcal{I}$ as follows.

- $\Delta^{\mathcal{I}} = \{d_C\} \cup \{d_A \mid A \in \mathsf{NE}(C)\} \cup \{d_a \mid a \in \mathsf{N_I}\}$;
- for each concept name A, $A^{\mathcal{I}} = \{d_X \mid X \sqsubseteq_{\mathcal{KB}} A\} \cup \{d_a \mid \mathcal{KB} \models A(a)\}$;
- for each role name R, we start by putting all edges that are classically required, i.e., $R^{\mathcal{I}} = \{(d_X, d_Y) \mid X \sqsubseteq_{\mathcal{KB}} \exists R.Y\} \cup \{(d_a, d_b) \mid R(a,b) \in \mathcal{KB}\} \cup \{(d_a, d_X) \mid \mathcal{KB} \models (\exists R.X)(a)\}$. Moreover, for each $\exists R.Y \in \mathsf{X}$, we add the edge (d_C, d_Y) to R^I. Extra edges starting from individuals other than d_C are not relevant.

By construction, $\mathcal{I}$ is a classical model of $\mathcal{KB}$ and, as $C' \notin \mathsf{X}$, $d_C \in \lfloor C \rfloor^{\mathcal{I}} \setminus C'^{\mathcal{I}}$. It remains to prove that there is no model $\mathcal{J}$ that improves $\mathcal{I}$ by making d_C satisfy $\hat{\delta}$.

If $\hat{\delta} \notin \mathsf{Comp}_{\mathsf{fix}}(C)$, then either $\mathsf{NE}(\exists R.A_2) \not\subseteq \mathsf{NE}(C)$ or there exists a concept name A' such that $A' \in \mathsf{SupCls}(\exists R.A_2)$ and $A' \notin \mathsf{SupCls}(C)$ (hence, $d_C \notin A'^{\mathcal{I}}$). Since any model $\mathcal{J}$ that is comparable with $\mathcal{I}$ has the same interpretation for the concept names, such model cannot have $d_C \in \exists(R.A_2)^{\mathcal{J}}$.

If instead $\hat{\delta} \notin \mathsf{Cons}(X')$, we have $\bigsqcap_{D' \in \mathsf{X}'} D' \sqcap \exists R.A_2 \sqsubseteq_{\mathcal{KB}} \bot$. If this inconsistency derives from classical consequences of C (i.e., $\exists R.A_2 \sqcap \mathsf{SupCls}(C) \sqsubseteq_{\mathcal{KB}} \bot$), the thesis is obvious. Otherwise, the inconsistency is due to one or more defeasible inclusions δ that were chosen in the previous iterations of the loop, on line 3. For each such δ, either its priority is higher than the one of $\hat{\delta}$, or it is incomparable with it. In the first case, clearly it is not worth modifying δ in order to improve $\hat{\delta}$. In the latter case, we employ the assumption that $\mathcal{KB}$ is conflict safe. In particular, we have that δ and $\hat{\delta}$ are incomparable and in conflict. Let $\delta = (A_3 \sqsubseteq_n \exists R.A_4)$. There is a concept name A_5 such that $A_5 \equiv_{\mathcal{KB}} A_1 \sqcap A_3$ and the defeasible inclusion $\delta' = (A_5 \sqsubseteq_n \exists R.A_4)$ belongs to $\mathcal{KB}$. Then, the priority of δ' is higher than both δ and $\hat{\delta}$. Hence, it is not worth modifying δ' to improve $\hat{\delta}$. □

Theorem 4. *Algorithm 1 runs in polynomial time.*

Proof. The main cycle of the algorithm performs as many iterations as the number of defeasible inclusions in $\mathcal{KB}$. The polynomial complexity of the auxiliary operators NE, SupCls, $\mathsf{Comp}_{\mathsf{fix}}$ and Cons derive from the polynomial complexity of reasoning in $\mathcal{EL}$. □

The following example shows how to apply Algorithm 1 to the KB of Example 2.

Example 5. Assume that $\mathcal{KB}$ is the knowledge base of Example 3 and we want to check whether staff members can read project files. First, we have to reduce the $\mathcal{KB}$ in normal form as follows. We introduce six new concept names — $\texttt{SubUsers}$, $\texttt{SubStaff}$, $\texttt{TargProjects}$, $\texttt{AuxUsers}$, $\texttt{AuxStaff}$ and $\texttt{ActRead}$ — together with the following equivalences.

$$\begin{aligned}
\exists\texttt{subject.Users} &\equiv \texttt{SubUsers}\\
\exists\texttt{subject.Staff} &\equiv \texttt{SubStaff}\\
\exists\texttt{target.Projects} &\equiv \texttt{TargProjects}\\
\exists\texttt{action.Read} &\equiv \texttt{ActRead}\\
\texttt{SubUsers} \sqcap \texttt{TargProjects} &\equiv \texttt{AuxUsers}\\
\texttt{SubStaff} \sqcap \texttt{TargProjects} &\equiv \texttt{AuxStaff}\\
\texttt{AuxUsers} \sqcap \texttt{ActRead} &\equiv \texttt{UserRequest}\\
\texttt{StaffUsers} \sqcap \texttt{ActRead} &\equiv \texttt{StaffRequest}
\end{aligned}$$

The above equivalences replace the original definitions of $\texttt{UserRequest}$ and $\texttt{StaffRequest}$. The other inclusions remain unchanged. Recall that the $\mathcal{KB}$ contains

$$\begin{aligned}
\top &\sqsubseteq \exists\texttt{aux.Grant}\\
\top &\sqsubseteq \exists\texttt{aux.Deny}
\end{aligned}$$

Algorithm 1 receives as input

$$C = \exists\texttt{subject.Staff} \sqcap \exists\texttt{target.Projects} \sqcap \exists\texttt{action.Read}.$$

On line 1, the superclasses of C are computed. At that point, X contains, among the others, $\texttt{StaffPolicy}$ and $\mathsf{NE}(C)$ contains $\texttt{Grant}$. According to specificity, the first defeasible inclusion removed from $\mathcal{D}$ is $\texttt{StaffPolicy} \sqsubseteq_{\mathtt{n}} \exists\texttt{decision.Grant}$. Since $\exists\texttt{decision.Grant}$ has no proper superclasses and $\mathsf{NE}(\exists\texttt{decision.Grant})$ contains only $\texttt{Grant}$, the condition on line 4 is satisfied and X becomes $X \cup \{\exists\texttt{decision.Grant}\}$. Thus, we have that

$$\begin{aligned}
&\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}}\\
&\exists\texttt{subject.Staff} \sqcap \exists\texttt{target.Projects} \sqcap \exists\texttt{action.Read} \sqsubseteq \exists\texttt{decision.Grant}.
\end{aligned}$$

Note that the second defeasible inclusion $\texttt{UsersPolicy} \sqsubseteq_{\mathtt{n}} \exists\texttt{decision.Deny}$ does not belong to $\mathsf{Cons}(X)$ since $\exists\texttt{decision.Grant}$ and $\exists\texttt{decision.Deny}$ are inconsistent. □

4 Reasoning about Individuals

The ideas illustrated so far can be naturally extended to reasoning about individuals, that is, instance checking. This task suffers from the same problem as

subsumption: given an assertion $A(a)$, the individual a might well be a member of any subclass of A, which may prevent the default properties of A from being inherited by a if the standard definition of instance checking [7] is used. Therefore, some form of closure similar to $CWA_{\mathcal{KB}}$ is needed. The closure, in this case, applies to the atomic concepts that contain the individuals in the ABox, as collected by the meta-function $AtCls_{\mathcal{KB}}(a) = \bigsqcap\{A \mid \mathcal{KB} \models A(a)\}$.

Definition 6. *Let $\mathcal{KB}$ be any defeasible $\mathcal{EL}^{\bot}$ KB. $CWA(\mathcal{KB})$ denotes the knowledge base obtained from $\mathcal{KB}$ by adding the assertions $CWA_{\mathcal{KB}}(AtCls_{\mathcal{KB}}(a))(a)$, for all individuals a occurring in $\mathcal{KB}$.*

Instance checking $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C(a)$ is then defined as $\mathsf{Circ}_{\mathsf{fix}}(CWA(\mathcal{KB})) \models C(a)$ or, in a model-theoretic view:

Definition 7. *$\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C(a)$ if and only if for all models $\mathcal{I}$ of $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB})$ if $\{A \in \mathsf{N_C} \mid a^{\mathcal{I}} \in A^{\mathcal{I}}\} = \{A \in \mathsf{N_C} \mid \mathcal{KB} \models A(a)\}$, then $a^{\mathcal{I}} \in C^{\mathcal{I}}$.*

Since $\mathsf{Circ}_{\mathsf{fix}}$ preserves the classical semantics of atomic concepts and $\mathcal{EL}^{\bot}$ KBs behave like Horn theories in many respects, it can be proved that:

Proposition 2. *For all defeasible $\mathcal{EL}^{\bot}$ knowledge bases $\mathcal{KB}$, and all conjunctions of atomic concepts C, $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C(a)$ iff $CWA(\mathcal{KB}) \models C(a)$ iff $\mathcal{KB} \models C(a)$.*

In other words, membership to atomic concepts and conjunctions thereof is fully classical. Therefore, in this paper, we focus on the more interesting problem of inferring the default properties of individuals. The reasoning task of our interest is the following: *Given an individual "a" and a concept $\exists R.A$, decide whether*

$$\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} (\exists R.A)(a) .$$

The NP-hardness proof for subsumption can be easily adapted to instance checking (using the same reduction plus assertion $B_0(a)$ and the query $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} (\exists \bar{F})(a)$). So we get:

Theorem 5. *Let $\mathcal{KB}$ range over $\mathcal{EL}^{\bot}$ knowledge bases. The problem of checking whether $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} C(a)$ is NP-hard, even if the existential restriction is unqualified (i.e., $A = \top$).*

For conflict safe knowledge bases, the instance checking problem can be decided using the same algorithm as for subsumption. What we need is to provide as input a concept which is the conjunction of all the atomic concepts and existential restrictions which a is classically an instance of. Let $GenCls_{\mathcal{KB}}(a)$ be such a conjunction:

$$GenCls_{\mathcal{KB}}(a) = \bigsqcap\{A \mid \mathcal{KB} \models A(a)\} \sqcap \bigsqcap\{\exists R.A \mid \mathcal{KB} \models (\exists R.A)(a)\} .$$

The proof of the following theorem is analogous to Theorem 3 and is left to the reader.

Theorem 6. *Let* X *be the result of Algorithm 1 on the concept* $GenCls_{\mathcal{KB}}(a)$*. If* $\mathcal{KB}$ *is conflict safe then* $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} (\exists R.A)(a)$ *iff* $(\exists R.A) \in \mathsf{X}$*.*

Example 6. Let $\mathcal{KB}$ be a knowledge base obtained by adding to Example 4 the assertions:

$$\texttt{Human}(\texttt{Mary})$$
$$\texttt{SitusInversus}(\texttt{John})$$

Recall that $\mathcal{KB}$ contains $\top \sqsubseteq \exists \texttt{aux}.\texttt{LHeart}$, where aux is a new role name.

We want to check that $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} (\exists \texttt{has\_heart}.\texttt{LHeart})(\texttt{Mary})$ and $\mathsf{Circ}_{\mathsf{fix}}(\mathcal{KB}) \models_{\mathsf{cw}} (\exists \texttt{has\_heart}.\texttt{RHeart})(\texttt{John})$. Let consider the first query, the input of Algorithm 1 is the concept $C = \texttt{Human}$. As Human has no proper superclasses, at line 1 $X = \{\texttt{Human}\}$. The only defeasible inclusion to be checked in lines 2-5 is $\texttt{Human} \sqsubseteq_n \exists \texttt{has\_heart}.\texttt{LHeart}$. The set $\mathsf{NE}(\texttt{Human})$ consists of all the concept names occurring in the knowledge base, $\exists \texttt{has\_heart}.\texttt{LHeart}$ is consistent with Human and it does not force other concept names to be locally true. Therefore, the condition in line 4 is satisfied and $\exists \texttt{has\_heart}.\texttt{LHeart}$ is added to X as expected.

For the second query, as seen before $\exists \texttt{has\_heart}.\texttt{RHeart}$ classically derives from SitusInversus and hence it is added to X directly in line 1. Note that, even if $\texttt{Human} \sqsubseteq_n \exists \texttt{has\_heart}.\texttt{LHeart}$ is *activated* by the fact that $\texttt{SitusInversus} \sqsubseteq_{\mathcal{KB}} \texttt{Human}$, the defeasible inclusion $\texttt{Human} \sqsubseteq_n \exists \texttt{has\_heart}.\texttt{LHeart}$ is not in $\mathsf{Cons}(X)$ because $\exists \texttt{has\_heart}.\texttt{LHeart}$ and $\exists \texttt{has\_heart}.\texttt{RHeart}$ are inconsistent. □

5 Related Work

DLs have been extended with nonmonotonic constructs such as default rules [20, 3, 4], autoepistemic operators [11, 12], and circumscription [10, 8, 7]. An advantage of circumscription is that nonmonotonic properties apply to all individuals, while the other approaches restrict nonmonotonic inferences to the individuals that are explicitly denoted in the ABox, as observed in [8]. While [8] focusses on expressive circumscribed description logics whose complexity may reach NEXPTIME$^{\text{NP}}$, [10] and [7] deal with lower-complexity DLs like $\mathcal{ALE}$, DL-lite, and $\mathcal{EL}$; however, upper complexity bounds are all at the second level of the polynomial hierarchy or harder, while here we have identified a tractable case. The two works [8, 7] consider more general forms of circumscription (with variable concept names) and reasoning tasks (satisfiability and KB consistency) that we do not consider here. However, they do not deal with the modified entailment $\models_{\mathsf{cw}}$ on which this paper is focussed. Another recent attempt at low-complexity, nonmonotonic DL reasoning is based on a modal *typicality operator* [15, 14], whose extension is maximized to achieve nonmonotonic inferences. Unfortunately, reasoning is intractable in this approach.

6 Conclusions and Perspectives

The need for supporting prototypical reasoning and exceptions in DLs can be addressed by restricting the expressiveness of the underlying DL and by select-

ing an appropriate form of inference ($\models_{\mathsf{cw}}$). We have shown how to encode a recurring example originated by the work on biomedical ontologies, and a representative example related to semantic web policies. The adoption of $\mathsf{Circ}_{\mathsf{fix}}$ makes it possible to add default attributes to the concepts of a given (classical) ontology in a controlled way, without affecting the extension of atomic concepts. For conflict safe KBs, the problem of reasoning about default attributes belongs to P; we provided an algorithm based on $\mathcal{EL}$ classification problems that enjoy efficient implementations [5]. This is a promising starting point for addressing the performance challenges posed by the semantic web.

In the full version of this paper we will provide more details on the strategies for making KBs conflict safe. We are also going to support more general queries and more constructs from $\mathcal{EL}^{++}$, identifying the tractability threshold.

An interesting direction for further research consists in studying the impact of variable concept names on the complexity of $\models_{\mathsf{cw}}$.

Acknowledgements. This work is partially supported by the national project LoDeN (`http://loden.fisica.unina.it/`). The authors are grateful to the anonymous referees for their constructive comments that helped improving the paper.

References

1. F. Baader. The instance problem and the most specific concept in the description logic EL w.r.t. terminological cycles with descriptive semantics. In *Proc. of the 26th Annual German Conference on AI, KI 2003*, volume 2821 of *Lecture Notes in Computer Science*, pages 64–78. Springer, 2003.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the EL envelope. In *Proc. of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI-05*, pages 364–369. Professional Book Center, 2005.
3. F. Baader and B. Hollunder. Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1):149–180, 1995.
4. F. Baader and B. Hollunder. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. Autom. Reasoning*, 15(1):41–68, 1995.
5. F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL - a polynomial-time reasoner for life science ontologies. In U. Furbach and N. Shankar, editors, *IJCAR*, volume 4130 of *Lecture Notes in Computer Science*, pages 287–291. Springer, 2006.
6. F. Baader, D. L. McGuiness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2003.
7. P. A. Bonatti, M. Faella, and L. Sauro. Defeasible inclusions in low-complexity DLs: Preliminary notes. In C. Boutilier, editor, *IJCAI*, pages 696–701, 2009.
8. P. A. Bonatti, C. Lutz, and F. Wolter. The complexity of circumscription in dls. *J. Artif. Intell. Res. (JAIR)*, 35:717–773, 2009.
9. P. A. Bonatti and P. Samarati. Logics for authorization and security. In J. Chomicki, R. van der Meyden, and G. Saake, editors, *Logics for Emerging Applications of Databases*, pages 277–323. Springer, 2003.

10. M. Cadoli, F. Donini, and M. Schaerf. Closed world reasoning in hybrid systems. In *Proc. of ISMIS'90*, pages 474–481. Elsevier, 1990.
11. F. M. Donini, D. Nardi, and R. Rosati. Autoepistemic description logics. In *IJCAI (1)*, pages 136–141, 1997.
12. F. M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225, 2002.
13. T. W. Finin, A. Joshi, L. Kagal, J. Niu, R. S. Sandhu, W. H. Winsborough, and B. M. Thuraisingham. ROWLBAC: representing role based access control in OWL. In I. Ray and N. Li, editors, *SACMAT*, pages 73–82. ACM, 2008.
14. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Prototypical reasoning with low complexity description logics: Preliminary results. In E. Erdem, F. Lin, and T. Schaub, editors, *LPNMR*, volume 5753 of *Lecture Notes in Computer Science*, pages 430–436. Springer, 2009.
15. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Reasoning about typicality in ALC and EL. In Grau et al. [16].
16. B. C. Grau, I. Horrocks, B. Motik, and U. Sattler, editors. *Proceedings of the DL Home 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.
17. V. Kolovski, J. A. Hendler, and B. Parsia. Analyzing web access control policies. In C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, editors, *WWW*, pages 677–686. ACM, 2007.
18. A. L. Rector. Defaults, context, and knowledge: Alternatives for OWL-indexed knowledge bases. In R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, pages 226–237. World Scientific, 2004.
19. R. Stevens, M. E. Aranguren, K. Wolstencroft, U. Sattler, N. Drummond, M. Horridge, and A. L. Rector. Using OWL to model biological knowledge. *International Journal of Man-Machine Studies*, 65(7):583–594, 2007.
20. U. Straccia. Default inheritance reasoning in hybrid KL-ONE-style logics. In *IJCAI*, pages 676–681, 1993.
21. A. Uszok, J. M. Bradshaw, R. Jeffers, N. Suri, P. J. Hayes, M. R. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. KAoS policy and domain services: Towards a description-logic approach to policy representation, deconfliction, and enforcement. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 93–96, Lake Como, Italy, June 2003. IEEE Computer Society.
22. R. Zhang, A. Artale, F. Giunchiglia, and B. Crispo. Using description logics in relation based access control. In Grau et al. [16].