# Towards Practical Defeasible Reasoning for Description Logics

Giovanni Casini, Thomas Meyer, Kody Moodley, and Ivan Varzinczak

Centre for Artificial Intelligence Research
CSIR Meraka Institute and UKZN, South Africa
Email: {GCasini,TMeyer,KMoodley,IVarzinczak}@csir.co.za

**Abstract.** The formalisation of defeasible reasoning in automated systems is becoming increasingly important. Description Logics (DLs) are nowadays the main logical formalism in the field of formal ontologies. Our focus in this paper is to devise a practical implementation for prior work that formalises a version of *Rational Closure* (an important type of defeasible reasoning) for DLs. We show that the conclusions drawn from it are generally intuitive and desirable. Moreover, we present experimental results showing that using Rational Closure for ontologies of reasonable size is practical.

## 1 Introduction

Reasoning with *exceptions* has been a major topic in AI since the 80's. The problem has been the assumption of certainty, in *monotonic* systems, of represented information when deriving inferences. These systems generally cannot accommodate the addition of new information which contradicts with what is known. For example, if a monotonic system is told that "*Students do not pay taxes*" then, upon encountering an exception (a student who works), it will still conclude that this student is exempt from taxes [14]. *Defeasible* reasoning is concerned with the development of formalisms which are able to represent and reason with defeasible (non-strict) facts:"Typically, *students do not pay taxes*" is the defeasible counterpart of "*Students do not pay taxes*".

The main approaches for introducing defeasible reasoning into KR formalisms (such as DLs [1]) have been through adaptations or combinations of the following systems: Circumscription [5, 29], Default Logic [27], Negation as failure [12, 20], Probabilistic logic [19, 24], Autoepistemic Logic [11] and Preferential reasoning [8, 9, 14].

The theoretical foundation of our work is a DL adaptation of the preferential reasoning approach by Lehmann et al. [22,23]. The motivation for focusing on the preferential approach is that it gives back intuitive inferences using procedures that reduce to classical DL reasoning. This gives the advantage of being able to use "off-the-shelf" DL reasoners to perform defeasible inference. Hinging on the decidability of classical DLs, we find that our preferential approach is also decidable. Here, we focus on a particular preferential construction, the *Rational Closure* [23], that has been adapted to the DL $\mathcal{ALC}$ [9].

In this paper, our goals are: to refine the practical implementation of the Rational Closure algorithm, based on the procedure presented in [9] (Section 2); to reiterate that, in a DL setting, Rational Closure makes intuitive and desirable inferences in general

(Section 3) and to show that it is practical to use Rational Closure in a DL setting from a performance perspective (Section 4). The latter performance evaluation is the main contribution of this work and is, to our knowledge, the first evaluation of this maturity and nature in the community. We assume that the reader is familiar with DLs [1] and $\mathcal{ALC}$ [28] in particular.

## 2 An algorithm to compute Rational Closure in $\mathcal{ALC}$

We present an algorithm for computing Rational Closure in $\mathcal{ALC}$, based on a procedure defined by Casini and Straccia [9] and similar in style to Pearl's System Z [26] and the possibilistic system by Benferhat et al. [2]; a version of this algorithm has been published in earlier work [25] and what we present here is a refinement to align fully with our semantic characterisation of Rational Closure [7]; we omit the theoretical underpinnings, referring the reader to Lehmann and Magidor's work [23] and, *w.r.t.* the DL reformulation, to the work of Casini and Straccia [9] and Britz et al. [7]. Rational Closure has a series of very desirable properties from the formal point of view: the consequence relation has a solid logical connotation, since it is characterised by a set of structural properties that should be satisfied by any non-monotonic formal system [7, 21, 23]; the decision problem can be reduced to a series of classical monotonic decision steps; eventually, the overall computational complexity of the procedure is the same as the one of the underlying entailment relation [7, 9].

In order to model defeasible reasoning in $\mathcal{ALC}$, we introduce a new kind of inclusion axiom, i.e., a *defeasible inclusion axiom* $C \precsim D$, which is read as "any object classified under the concept $C$ is *typically* also classified under the concept $D$", that is, if we are informed that an object is in the set referred to by $C$ we can conclude that it is also in the set referred to by $D$, provided we do not have any other information forcing us to conclude otherwise. For the semantics of such axioms, we refer the reader to the work by Britz et al. [7, 8].

We consider knowledge bases (KBs) of the form $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$, where $\mathcal{T}$ is a DL TBox and $\mathcal{D}$ is known as a *DBox* which is a finite set of defeasible inclusion axioms. We are not considering the ABox here, and the algorithm we are going to introduce computes the Rational Closure only considering concepts, i.e., it will consider KBs $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ and it will be able to decide if $C \precsim D$ or $C \sqsubseteq D$ is in the Rational Closure of $\mathcal{K}$ (i.e., if it is a defeasible consequence of $\mathcal{K}$ according to Rational Closure).

To be more specific, we shall consider only KBs composed by a DBox $\mathcal{D}$: since from the point of view of the procedure the two axioms $C \sqsubseteq \bot$ and $C \precsim \bot$ are equivalent and each classical inclusion axiom $C \sqsubseteq D$ is equivalent to the defeasible inclusion axiom $C \sqcap \neg D \precsim \bot$, as explained by Casini and Straccia [9]. Hence it is always possible to transform a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$ into an equivalent KB $\mathcal{K}' = \langle \emptyset, \mathcal{D}^{\mathcal{K}} \rangle$, where $\mathcal{D}^{\mathcal{K}} = \mathcal{D} \cup \{C \sqcap \neg D \precsim \bot \mid C \sqsubseteq D \in \mathcal{T}\}$.

*Example 1.* Consider the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$, with $\mathcal{T} = \{\mathsf{BactMen} \sqsubseteq \mathsf{Men}, \mathsf{VirMen} \sqsubseteq \mathsf{Men}\}$ and $\mathcal{D} = \{\mathsf{Men} \precsim \neg\mathsf{Fatal}, \mathsf{BactMen} \precsim \mathsf{Fatal}\}$. $\mathcal{K}$ is about meningitis ($\mathsf{Men}$), bacterial meningitis ($\mathsf{BactMen}$), viral meningitis ($\mathsf{VirMen}$), and their fatality ($\mathsf{Fatal}$):

We can transform it into a KB composed of just a DBox $\mathcal{D}^{\mathcal{K}} = \{\mathsf{BactMen} \sqcap \neg\mathsf{Men} \precsim \bot, \mathsf{VirMen} \sqcap \neg\mathsf{Men} \precsim \bot, \mathsf{Men} \precsim \neg\mathsf{Fatal}, \mathsf{BactMen} \precsim \mathsf{Fatal}\}$.

From now on, when talking about a KB $\mathcal{K}$, we assume that all the information in the TBox has been moved into the DBox, i.e., we shall assume that we are talking about a KB $\langle \emptyset, \mathcal{D}^{\mathcal{K}} \rangle$. The use of KBs of the form $\langle \mathcal{T}, \mathcal{D} \rangle$ will be functional to the exposition, and has to be considered simply as a renaming of the correspondent DBox $\mathcal{D}^{\mathcal{K}}$.

If all the axioms in $\mathcal{K}$ were classical inclusion axioms, we would derive that bacterial meningitis is at the same time fatal ($\mathsf{BactMen} \sqsubseteq \mathsf{Fatal}$) and non-fatal ($\mathsf{BactMen} \sqsubseteq \mathsf{Men}$, $\mathsf{Men} \sqsubseteq \neg\mathsf{Fatal}$), i.e., it would have turned out to be an empty concept. Thus, we rather represent some of the strict facts in $\mathcal{K}$as defeasible ones to exhibit the atypicality of $\mathsf{BactMen}$ ($\mathsf{BactMen}$ is atypical w.r.t. to its parent class $\mathsf{Men}$ because it has a property $\mathsf{Fatal}$ which is in direct contradiction to a property of its parent i.e. $\neg\mathsf{Fatal}$).

We shall indicate with $\overline{\mathcal{D}}$ the sets of the *materialisations* of the axioms in $\mathcal{D}$, where the *materialisation* of an axiom $C \sqsubseteq\!\!\!\!\sim D$ denotes the concept expressing the same subsumption relation of the axiom (i.e., $\neg C \sqcup D$). Hence $\overline{\mathcal{D}} = \{\neg C \sqcup D \mid C \sqsubseteq\!\!\!\!\sim D \in \mathcal{D}\}$.

The algorithm to compute Rational Closure consists of a main algorithm and two sub-procedures. All three are based solely on classical entailment for $\mathcal{ALC}$ ($\models$). The first sub-procedure is called *exceptional*. Its aim is to determine which of the concepts named in our axioms are exceptional. Intuitively, a concept is exceptional in a KB if it refers to a class that is atypical w.r.t. one of its superclasses (*e.g.* $\mathsf{BactMen}$ in the example above). Technically, the exceptionality of a concept can be decided using $\models$, since a concept $C$ is exceptional in $\mathcal{K} = \langle \emptyset, \mathcal{D} \rangle$ if and only if $\models \bigsqcap \overline{\mathcal{D}} \sqsubseteq \neg C$.[1] A defeasible axiom $C \sqsubseteq\!\!\!\!\sim D \in \mathcal{D}$ is considered exceptional if its antecedent $C$ is exceptional. Given a finite set $\mathcal{E}$ of defeasible inclusion axioms, the algorithm *exceptional* gives back the subset of $\mathcal{E}$ containing the exceptional axioms.

**Procedure** $exceptional(\mathcal{E})$

```
  Input: $\mathcal{E} \subseteq \mathcal{D}$
  Output: $\mathcal{E}' \subseteq \mathcal{E}$ such that $\mathcal{E}'$ is exceptional w.r.t. $\mathcal{E}$
1 $\mathcal{E}' := \emptyset$;
2 foreach $C \sqsubseteq\!\!\!\!\sim D \in \mathcal{E}$ do
3     if $\models \bigsqcap \overline{\mathcal{E}} \sqsubseteq \neg C$ then
4         $\mathcal{E}' := \mathcal{E}' \cup \{C \sqsubseteq\!\!\!\!\sim D\}$;
5 return $\mathcal{E}'$;
```

Once we have defined the notion of exceptionality, we can think of ordering all the defeasible axioms in our KB with respect to their exceptionality. That is, we can associate a ranking value to each axiom in the KB by a recursive application of the algorithm *exceptional*. *computeRanking* is the algorithm that, using *exceptional* as a sub-procedure, partitions the set $\mathcal{D}$ into $\mathcal{R} = \{\mathcal{D}_0, \mathcal{D}_1, \ldots\}$, where each set $\mathcal{D}_i$ contains the defeasible axioms having $i$ as ranking value.

[1] This is the only difference between the present procedure and the ones presented by Casini and Straccia [9] and Moodley et al. [25], where a concept $C$ is considered exceptional in $\mathcal{K}$ if and only if $\mathcal{K} \models \top \sqsubseteq \neg C$; the need for such a change has become apparent once we have defined an appropriate semantics for the Rational Closure in $\mathcal{ALC}$ [7].

**Procedure** $computeRanking(\mathcal{K})$

```
Input: Defeasible KB $\mathcal{D}$
Output: The ranking $\mathcal{R}$ for $\mathcal{D}$
1 $\mathcal{E}_0 := \mathcal{D}$; $\mathcal{E}_1 :=$ exceptional($\mathcal{E}_0$); $i := 0$;
2 while $\mathcal{E}_{i+1} \neq \mathcal{E}_i$ do
3     $i := i + 1$; $\mathcal{E}_{i+1} :=$ exceptional($\mathcal{E}_i$);
4 $\mathcal{D}_\infty := \mathcal{E}_i$; $\mathcal{R} := \{\mathcal{D}_\infty\}$;
5 for $j = 1$ to $i$ do
6     $\mathcal{D}_{j-1} := \mathcal{E}_{j-1} \backslash \mathcal{E}_j$; $\mathcal{R} := \mathcal{R} \cup \{\mathcal{D}_{j-1}\}$;
7 return $\mathcal{R}$;
```

*computeRanking* receives as input our DBox $\mathcal{D}$. It starts identifying the exceptional axioms in $\mathcal{D}$ (i.e., the set $\mathcal{E}_1$), then the exceptional axioms in $\mathcal{E}_1$ (i.e., the set $\mathcal{E}_2$), and so on. For some $i$, it will necessarily turn out that $\mathcal{E}_i = \mathcal{E}_i + 1$: that means that $\mathcal{E}_i$ (possibly being empty) is a fixed point of *exceptional*, and we say that the axioms in $\mathcal{E}_i$ have $\infty$ as ranking value. That implies that we derive the negation of the antecedents of such axioms at every step of the ranking construction. That is, we cannot conceive of a situation so exceptional such that such concepts are non-empty. That means that the negation of such antecedents is not properly defeasible information, and can be treated as strict information: $C \precsim D$ being in $\mathcal{D}_\infty$ is equivalent to saying that $C \sqsubseteq \bot$ is in our KB. Note that any axiom $C \sqcap \neg D \precsim \bot$ obtained from a classical inclusion axiom $C \sqsubseteq D$ always turns out to have $\infty$ as a ranking value, which corresponds to $C \sqcap \neg D \sqsubseteq \bot$, which in turn is logically equivalent to the original $C \sqsubseteq D$. Hence $\mathcal{D}_\infty$ will contain all the information in the original $\mathcal{T}$ plus, possibly, some non-defeasible information that implicitly follows from the defeasible axioms. We give an example to illustrate this:

*Example 2.* Consider a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{D} \rangle$, with $\mathcal{T} = \{E \sqsubseteq D\}$ and $\mathcal{D} = \{C \precsim \neg D, C \precsim E\}$. $\mathcal{K}$ is transformed into $\mathcal{D}^{\mathcal{K}} = \{C \precsim \neg D, C \precsim E, E \sqcap \neg D \precsim \bot\}$. We apply the ranking procedure, and we obtain that both $E \sqcap \neg D$ and $C$ are exceptional concepts, i.e., $\mathcal{E}_1 = \mathcal{E}_0 = \mathcal{D}^{\mathcal{K}}$. Such a result implies that $\mathcal{D}_\infty = \mathcal{D}^{\mathcal{K}}$, and therefore, that our initial KB $\mathcal{K}$ is equivalent to the TBox $\mathcal{T}' = \{E \sqsubseteq D, C \sqsubseteq \neg D, C \sqsubseteq E\}$ because we obtain the same ranking for both. Note that the information that $C \sqsubseteq \bot$ was not explicit in the original KB $\mathcal{K}$, and only became derivable when additionally considering the DBox.

Once the algorithm has identified $\mathcal{D}_\infty$, it ranks all the other axioms: an axiom has $i$ as a ranking value if $i$ is the highest label for which it turns out to be exceptional, that is, if it is in $\mathcal{E}_i$ but it does not appear anymore in $\mathcal{E}_{i+1}$. We obtain a partition of $\mathcal{D}$ into $\mathcal{R} = \{\mathcal{D}_0, \dots, \mathcal{D}_{i-1}, \mathcal{D}_\infty\}$.

*Example 3.* Consider the KB in Example 1. $computeRanking$ takes $\mathcal{D}^{\mathcal{K}}$ as input and begins to compute the ranking. The result of Lines 1 to 3 will be the sequence $\mathcal{E}_0 = \mathcal{D}^{\mathcal{K}}$, $\mathcal{E}_1 = \{\mathsf{BactMen} \sqcap \neg\mathsf{Men} \precsim \bot, \mathsf{VirMen} \sqcap \neg\mathsf{Men} \precsim \bot, \mathsf{BactMen} \precsim \mathsf{Fatal}\}$, $\mathcal{E}_2 = \mathcal{E}_3 = \{\mathsf{BactMen} \sqcap \neg\mathsf{Men} \precsim \bot, \mathsf{VirMen} \sqcap \neg\mathsf{Men} \precsim \bot\}$. Hence, by the instructions from Line 4 to Line 6, we obtain a partition of $\mathcal{D}^{\mathcal{K}}$ in $\mathcal{D}_0^{\mathcal{K}} = \{\mathsf{Men} \precsim \neg\mathsf{Fatal}\}$, $\mathcal{D}_1^{\mathcal{K}} = \{\mathsf{BactMen} \precsim \mathsf{Fatal}\}$, and $\mathcal{D}_\infty^{\mathcal{K}} = \{\mathsf{BactMen} \sqcap \neg\mathsf{Men} \precsim \bot, \mathsf{VirMen} \sqcap \neg\mathsf{Men} \precsim \bot\}$.

Now, given a defeasible KB $\mathcal{K} = \{\emptyset, \mathcal{D}\}$, we can obtain a ranking $\mathcal{R} = \{\mathcal{D}_0, \ldots, \mathcal{D}_n, \mathcal{D}_\infty\}$. Once this ranking is identified we can ask a query of the form $C \precsim D$ (noting that we can translate strict queries of the form $C \sqsubseteq D$ into $C \sqcap \neg D \precsim \bot$). Algorithm 1 can determine whether such queries are in the Rational Closure of $\mathcal{K}$. Note that if we are confronted with a strict query (classical inclusion axiom $C \sqsubseteq D$), one can determine if it is in the Rational Closure of the KB by checking if it is classically entailed by the strict information in $\mathcal{D}$ ($\mathcal{D}_\infty$). This can be implemented as an optimisation.

**Algorithm 1:** Rational Closure

```
   Input: The ranking $\mathcal{R}$ of $\mathcal{D}$ and query $\varphi$
   Output: true iff $\varphi$ is in the Rational Closure of $\mathcal{K}$
1  $n := 0$; $\mathcal{D}_\mathcal{R} := \mathcal{D} \backslash \mathcal{D}_\infty$;
2  while $\models \bigsqcap \overline{\mathcal{D}_\infty} \sqcap \bigsqcap \overline{\mathcal{D}_\mathcal{R}} \sqsubseteq \neg C$ and $\mathcal{D}_\mathcal{R} \neq \emptyset$ do
3      $\mathcal{D}_\mathcal{R} := \mathcal{D}_\mathcal{R} \backslash \mathcal{D}_n$; $n := n + 1$;
4  return $\models \bigsqcap \overline{\mathcal{D}_\infty} \sqcap \bigsqcap \overline{\mathcal{D}_\mathcal{R}} \sqcap C \sqsubseteq D$;
```

However, for simplicity, Algorithm 1 considers only the case in which the query is a defeasible inclusion axiom. The algorithm takes as input the ranking and query $C \precsim D$ and determines which portion of $\mathcal{R}$ is compatible with the concept $C$, i.e., which portion of defeasible information does not imply the negation of $C$, starting from the most normal situations up to increasing levels of exceptionality. We give an example:

*Example 4.* Consider the ranking $\mathcal{R}$ in Example 3 and the query $\varphi = \mathsf{VirMen} \precsim \neg\mathsf{Fatal}$ which we pose to Algorithm 1. The algorithm checks if $\models \bigsqcap \overline{\mathcal{D}^\mathcal{K}} \sqsubseteq \neg\mathsf{VirMen}$, which is not the case. Hence, we have to check if $\models \bigsqcap \overline{\mathcal{D}_\infty} \sqcap \bigsqcap \overline{\mathcal{D}_\mathcal{R}} \sqcap \mathsf{VirMen} \sqsubseteq \neg\mathsf{Fatal}$, which is true. On the other hand, if our query is $\varphi = \mathsf{BactMen} \precsim \neg\mathsf{Fatal}$, we obtain a different result: since $\models \bigsqcap \overline{\mathcal{D}^\mathcal{K}} \sqsubseteq \neg\mathsf{BactMen}$ but $\not\models \bigsqcap \overline{\mathcal{D}^\mathcal{K}/\mathcal{D}_0} \sqsubseteq \neg\mathsf{BactMen}$, $\mathsf{BactMen}$ is an exceptional class of level 1, compatible with $\mathcal{D}_1$, and we have to increase the exceptionality level eliminating the information in $\mathcal{D}_0$ from $\mathcal{D}_\mathcal{R}$. It turns out that $\not\models \bigsqcap \overline{\mathcal{D}_\infty} \sqcap \bigsqcap \overline{\mathcal{D}_\mathcal{R}} \sqcap \mathsf{BactMen} \sqsubseteq \neg\mathsf{Fatal}$, that is the right conclusion since we have $\mathsf{BactMen} \precsim \mathsf{Fatal}$ in our KB.

The computational complexity of the entire procedure is the same as that of the underlying monotonic entailment relation $\models$, i.e., it is an EXPTIME-complete problem ( [7] and [9], Corollary 2). This is easy to see since the number of classical entailment checks is at most exponential w.r.t. the size of the ontology (number of axioms). Moreover, note that the defined procedures can be applied to all the DLs that are more expressive than $\mathcal{ALC}$, still preserving the computational complexity of the decision problem w.r.t. the underlying monotonic entailment relation. Using a more expressive DL than $\mathcal{ALC}$, the defeasible information will be still represented only by defeasible inclusion axioms $C \precsim D$, while the strict information different from $\mathcal{ALC}$ inclusion axioms (role inclusion axioms, role transitivity, etc.) must be considered as background knowledge at each step of the decision procedure. The correctness of Algorithm 1 follows from the procedure by Casini and Staccia [9] as it is a direct translation/rewriting thereof.

## 3 Test Suite

We would like to argue that the kind of reasoning that Rational Closure models is satisfying from an intuitive point of view, i.e., that the conclusions we can draw are reasonable. For a thorough semantical motivation of this the reader should consult our theoretical work [7]. In this section we make use of a test suite consisting of ontology snippets widely used in the community to show that our presented procedure gives back the same desirable inferences that some other non-monotonic formalisms are able to derive as well as some which they are not able to derive.

**Eukaryotic Cells** [30]. Eukaryotic cells ($\mathsf{EukCell}$) have a proper nucleus, but there are some cells lacking a proper nucleus and are nonetheless considered eukaryotic, such as the mammalian red blood cells ($\mathsf{MamRedBloodCell}$).

$$\mathcal{K} = \left\{ \begin{array}{l} \mathsf{EukCell} \precsim \exists \mathsf{hasNucleus}.\top, \\ \mathsf{MamRedBloodCell} \sqsubseteq \mathsf{EukCell} \sqcap \neg\exists \mathsf{hasNucleus}.\top \end{array} \right\}$$

Using only classical subsumption, would imply the non-existence of mammalian red blood cells ($\mathsf{MamRedBloodCell} \sqsubseteq \bot$), while using defeasible subsumption we obtain the ranking $\mathcal{D}_0^{\mathcal{K}} = \{\mathsf{EukCell} \precsim \exists \mathsf{hasNucleus}.\top\}$ and $\mathcal{D}_\infty^{\mathcal{K}} = \{\mathsf{MamRedBloodCell} \sqcap \neg(\mathsf{EukCell} \sqcap \neg\exists \mathsf{hasNucleus}.\top) \precsim \bot\}$ that allows mammalian red blood cells to exist as exceptional eukaryotic cells, without a nucleus. The query $\mathsf{MamRedBloodCell} \precsim \exists hasNucleus.\top$ returns a negative answer, since the concept $\mathsf{MamRedBloodCell}$ can be associated only with $\mathcal{D}_\infty^{\mathcal{K}}$ and not with the entire $\mathcal{D}^{\mathcal{K}}$. If we take under consideration another kind of eukaryotic cell, *e.g.* the cells composing the muscle of a mammal (we add to our KB $\mathsf{MamMuscCell} \sqsubseteq \mathsf{EukCell}$), in the absence of more information the procedure associates the concept $\mathsf{MamMuscCell}$ with all the defeasible information $\mathcal{D}^{\mathcal{K}}$, concluding that it is a typical eukaryotic cell, and hence it has a nucleus ($\mathsf{MamMuscCell} \precsim \exists \mathsf{hasNucleus}.\top$).

Other well-known test-examples that present the same structure (a subclass that does not satisfy some properties typically characterising a super-class) are the *Situs Inversus* and the *Whale* examples [6], and the present procedure treats them in the same way.

**Access Control** [6]. In this example we deal with an extra level of exceptionality. A user typically does not have access to a confidential file, but members of the staff do. However, if a staff member is black listed, the access is revoked.

$$\mathcal{K} = \left\{ \begin{array}{l} \mathsf{User} \precsim \neg\exists \mathsf{AccessTo}.\mathsf{Confidential}, \\ \mathsf{Staff} \precsim \exists \mathsf{AccessTo}.\mathsf{Confidential}, \ \ \mathsf{Staff} \sqsubseteq \mathsf{User}, \\ \mathsf{BlackListedStaff} \sqsubseteq \mathsf{Staff} \sqcap \neg\exists \mathsf{AccessTo}.\mathsf{Confidential} \end{array} \right\}$$

If we used only classical subsumption, we would have derived $\mathsf{Staff} \sqsubseteq \bot$ and $\mathsf{BlackListedStaff} \sqsubseteq \bot$, that would have allowed for undesired conclusions (e.g., $\mathsf{Staff} \sqsubseteq \neg\exists \mathsf{AccessTo}.\mathsf{Confidential}$ and $\mathsf{BlackListedStaff} \sqsubseteq \exists \mathsf{AccessTo}.\mathsf{Confidential}$). Instead, using $\precsim$ and Procedure $computeRanking$ we end up with the ranking: $\mathcal{D}_0^{\mathcal{K}} = \{\mathsf{User} \precsim \neg\exists AccessTo.Confidential\}$, $\mathcal{D}_1^{\mathcal{K}} = \{\mathsf{Staff} \precsim \exists \mathsf{AccessTo}.\mathsf{Confidential}\}$, and $\mathcal{D}_\infty^{\mathcal{K}} = \{\mathsf{Staff} \sqcap \neg\mathsf{User} \precsim \bot, \mathsf{BlackListedStaff} \sqcap \neg(\mathsf{Staff} \sqcap \neg\exists \mathsf{AccessTo}.\mathsf{Confidential}) \precsim \bot\}$. Hence the concept $\mathsf{Staff}$ can be associated only with the default information in $\mathcal{D}_1^{\mathcal{K}} \cup \mathcal{D}_\infty^{\mathcal{K}}$, avoiding the conclusion $\mathsf{Staff} \precsim \neg\exists \mathsf{AccessTo}.\mathsf{Confidential}$. In the same way, the concept $\mathsf{BlackListedStaff}$ can be associated only with $\mathcal{D}^{\mathcal{K}}_{\infty}$, avoiding a positive answer to the query $\mathsf{BlackListedStaff} \sqsubset\!\!\!\sim \exists \mathsf{AccessTo.Confidential}$.

**Bee Key.** In this example we go beyond $\mathcal{ALC}$, since we make use of qualified number restrictions. It is a real-world example of classification/taxonomisation of bee species in sub-Saharan Africa [13] into genera and subgenera. The problem is that there are exceptions in the characteristics of bee species, and this makes species difficult to classify into genera and hence genera into families. A case in point is that most male afrotropical bees have thirteen segment antennae ($\mathsf{SegAnt}$). An afrotropical male belonging to sub-family apidae ($\mathsf{ApiFBee}$) and the pasite genus ($\mathsf{PGBee}$), however, only has twelve segment antennae. We can formalise such information as:

$$\mathcal{K} = \left\{ \begin{array}{l} \mathsf{PGBee} \sqcap \mathsf{Male} \sqsubseteq \; = 12\ \mathsf{hasPart.SegAnt}, \\ \mathsf{PGBee} \sqsubseteq \mathsf{ApiFBee}, \;\; \mathsf{ApiFBee} \sqsubseteq \mathsf{AfroSFBee}, \\ \mathsf{AfroSFBee} \sqcap \mathsf{Male} \sqsubset\!\!\!\sim \; = 13\ \mathsf{hasPart.SegAnt} \end{array} \right\}$$

Using a defeasible axiom, we avoid the concept $\mathsf{PGBee} \sqcap \mathsf{Male}$ to turn out as necessarily empty (i.e., the procedure gives back a negative answer to the query $\mathsf{PGBee} \sqcap \mathsf{Male} \sqsubset\!\!\!\sim \; = 13\ \mathsf{hasPart.SegAnt}$), while keeping the information that typically males of the afrotropical family have thirteen-segment antennas. We have shown that Rational Closure derives intuitive conclusions for our test suite. It must be noted, however, that in some cases the inferential power of Rational Closure turns out to be insufficient. That is, there may be conclusions that are intuitively desirable, but we are not able to derive. Once a class turns out to be atypical *w.r.t.* one of its superclasses, it cannot inherit *any* of the typical properties associated to each of its super-classes. For example, if we consider the Access Control case above, and add to the KB the axiom $\mathsf{User} \sqsubset\!\!\!\sim \exists \mathsf{AccessTo.Public}$, i.e., every user has usually access to public files. Applying the ranking procedure to the new KB, we obtain the same results as above, only with the addition of $\mathsf{User} \sqsubset\!\!\!\sim \exists \mathsf{AccessTo.Public}$ in $\mathcal{D}^{\mathcal{K}}_{0}$. As above, the axioms in $\mathcal{D}^{\mathcal{K}}_{0}$ cannot be associated to exceptional concepts as $\mathsf{Staff}$ and $\mathsf{BlackListedStaff}$, hence we cannot derive neither $\mathsf{Staff} \sqsubset\!\!\!\sim \exists \mathsf{AccessTo.Public}$ nor $\mathsf{BlackListedStaff} \sqsubset\!\!\!\sim \exists \mathsf{AccessTo.Public}$, that would have been desirable conclusions. In order to overcome such inferential limits, we have to extend the inferential power of Rational Closure. Among the proposed extensions, the most well-known one is the Lexicographic Closure [10, 22] which addresses the shortcoming of Rational Closure described above. We plan to present the procedure for computing it in DLs and the relevant experimental results in future publications.

## 4 Experiments

We have run preliminary experiments to determine the practical performance of the Rational Closure algorithm. To our knowledge there are no other published evaluations of this scale or nature for defeasible reasoning approaches. In this section we give a description of the data we generated and present the results of our experiments.

**Setup.** We initially considered using, as our data, $\mathcal{EL}^{\perp}$ test ontologies graciously made available on the LoDEN (http://loden.fisica.unina.it) project website. LoDEN stands for Low complexity Description Logics with Non-monotonic features. In the end, although the ontologies were of sufficient size, we decided that there were too few in the dataset to be statistically significant. Also, we decided that we are more interested in the performance for $\mathcal{ALC}$. Therefore, we generated our own dataset as follows to address our needs: we randomly generated 11 sets of 50 OWL ontologies each (all represented in $\mathcal{ALC}$). Each set represented a different *percentage defeasibility* (ratio of defeasible to strict axioms in the ontologies) in increments of 10 from 0 to 100. The number of axioms in the ontologies of each set varied uniformly between 150 and 5150.

It is notable that the size of the ontologies cannot be considered large-scale to be representative of mature bio-medical ontologies such as those stored in the NCBO BioPortal corpus (http://bioportal.bioontology.org) but at the same time our ontologies are by no means small and give an accurate reflection of the average sizes of application ontologies in other corpuses such as the SWEET (http://sweet.jpl.nasa.gov) corpus. We motivate the use of randomly generated ontologies by observing that the defeasibility paradigm is not as yet widely embraced in "real-world" ontology development. There simply aren't any applied ontologies in existence incorporating similar notions of defeasibility to which we are presenting. The idea of these experiments was to get an initial sense for how well the Rational Closure algorithm performs with non-trivial ontologies of reasonable size. We conjecture that the data we have generated is qualitatively and quantitatively appropriate as a first attempt to determine this. Our $\mathcal{ALC}$ ontology generator and test ontologies are available for download at: http://tinyurl.com/onwddh6.

In addition to the ontologies, we also randomly generated a set of TBox queries for each ontology using terms in their signatures (concept and role names in the ontology). The total number of queries generated per ontology was 2 percent of the number of axioms in that ontology. The task was then to check entailment of the queries in each ontology using Rational Closure (Section 2). The first step was to generate the rankings of the ontologies; and then finally to execute the queries. We recorded the average ranking computation times, the number of ranks that occurred in each ranking and the average query answering times. The experiments were carried out on an Intel Core 2 Duo machine with 2GB of memory allocated to the JVM (Java virtual machine). The classical DL reasoning implementation used was HermiT (http://www.hermit-reasoner.com) accessed through the OWLAPI (http://owlapi.sourceforge.net).

**Results.** From the perspective of ontology engineering, we view the computation of the ranking of a certain version of an ontology as a task to be executed offline. The typical scenario is that the ranking will be computed once a stable version of the ontology is obtained and then stored offline. When query answering needs to be carried out the ranking can be loaded on demand. The average number of ranks per ranking of an ontology that we encountered in our dataset is shown in Figure 1.

In terms of the average performance of the ranking computation procedure, our results show a steady increase in this measurement as the percentage defeasibility of ontologies increases. In the case of 10 percent defeasiblility we obtained an average ranking computation time of less than half a second. In the worst case, that is in the case of ontologies with 100 percent defeasibility the average time to compute a ranking was found to be in the region of 8 seconds. In the special case where we have 0 percent defeasibility, query answering reduces to classical DL entailment and hence no ranking is required. If we study the impact of the number of ranks in a particular ranking on our performance results, we find that we obtained 30 different values for the number of

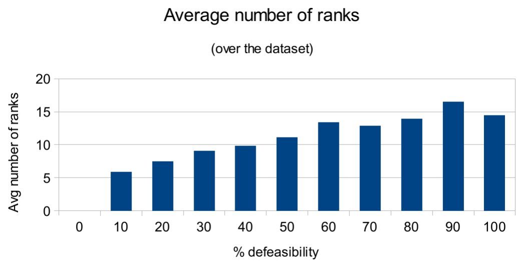

**Fig. 1.** Average number of ranks in a ranking over the ontologies in the dataset.

ranks in our dataset ranging between 0 and 49. If we disregard the 0 defeasibility case, the difference between the times for computing the rankings with the lowest number of ranks (3) and the highest number of ranks (49) is about 22.5 seconds. Results are depicted in Figure 2.

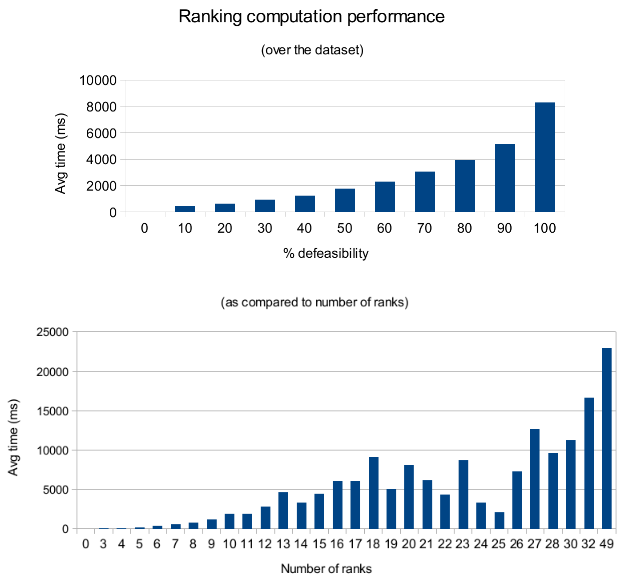

**Fig. 2.** The top graph indicates the average time taken to compute the ranking for an ontology in each category of defeasibility. The bottom graph indicates the average time taken to compute the ranking for an ontology in the dataset that has the indicated number of ranks.

In summary, for computation of the ranking of the ontologies in our dataset we can conclude that on average computing the ranking takes between 0.4 and 8 seconds for ontologies that are between 150 and 5150 axioms large. This seems to indicate that computation of the ranking, as an offline task, is feasible from an ontology engineering point of view. It is also clear from the results that the number of ranks in the ranking affects the performance of the ranking computation. It is likely that this is because of more entailment checks required to compute rankings with more ranks. We note that updating of ontologies to new versions will require recomputation of the ranking although this need not be in a naive way. We have optimisations in the pipeline for modifying existing rankings to reflect the new changes in the ontology.

In terms of query execution, our results show that defeasible reasoning is practical in ontologies of similar sizes to those in our dataset. With an additional optimisation step we have pruned away axioms in the ranking that are irrelevant to the signature of the query using ontology modularisation techniques [17, 18]. The results are that executing a single TBox query takes on average between 1.2 and 1.7 milliseconds depending on the percentage of defeasibility of the ontologies. We note also that defeasible reasoning is very close to the performance of classical entailment (0.7 milliseconds in the 0 percent defeasible case). These results strongly indicate that query answering can be executed on demand at least in the context of average ontology sizes between 150 and 5150 and average ranking sizes up to 50. Query results are shown in Figure 3.

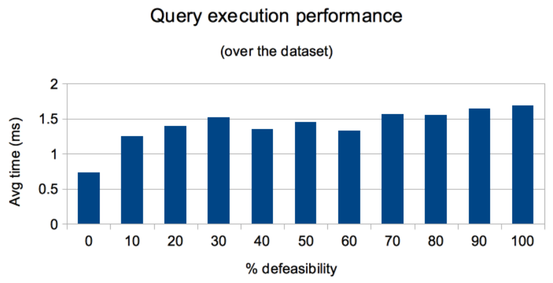

**Fig. 3.** The average time taken to execute a query for an ontology in each category of defeasibility.

## 5 Related Work

Closely related to our work from a theoretical point of view is that of Giordano et al. [15, 16] which uses preferential orderings on the individuals to define a typicality operator $\mathsf{T}$ *s.t.* the expression $\mathsf{T}(C) \sqsubseteq D$ corresponds to our $C \precsim D$. They provide a tableau calculus for their system that relies on the properties of the preferential consequence relations. In order to augment the inferential power of their system they have used circumscription techniques, obtaining systems that share properties of both the preferential and the circumscriptive approaches. At present, we are not aware of any implementation of their work. Outside the family of preferential systems there are mature proposals based on circumscription for DLs by Bonatti et al. [3,4,6] and by Sengupta et al. [29]. To our knowledge, the proposal by Bonatti et al. is the only one, together with the present one, that has been properly implemented. We are aware that they have performance results for their latest circumscriptive approach which are not yet published to our knowledge, and we hereby acknowledge their input to our work through sharing of test data for comparison purposes.

The main drawback of circumscriptive approaches is the burden on the ontology engineer to make appropriate decisions related to the fixing and varying of concepts and the priority of defeasible subsumption statements. Such choices can have a major effect on the conclusions drawn from the system, and can easily lead to counter-intuitive inferences. For example, in the Eukaryotic Cell example, circumscription can derive the conclusion $\mathsf{MamRedBloodCell} \sqsubseteq \bot$ (Bonatti's personal communication), which means that the existence of mammalian red blood cells is impossible, that is clearly an undesirable result. Moreover, the use of circumscription usually implies a considerable increase in computational complexity *w.r.t.* the underlying monotonic entailment relation (TBox reasoning with concept-circumscribed KBs for $\mathcal{ALC}$ is $\text{NEXP}^{\text{NP}}$-complete [6]), in contrast to the EXPTIME-complete complexity of our approach (Section 2).

## 6 Conclusions and future work

We have presented an algorithm for computing Rational Closure for $\mathcal{ALC}$ (Section 2), and we have shown that such a procedure gives back desirable conclusions *w.r.t.* some representative test-examples extracted from the literature (Section 3). The main contribution of the present work is a first but significant practical evaluation of the performance of Algorithm 1 *w.r.t.* ontologies of moderate size. Our results show that query answering can be computed on demand and is roughly twice as slow as classical entailment over the dataset. We have seen that despite the fact that computation of classical entailment is clearly more efficient, the cost of the implementation of Rational Closure is not at all excessive. Therefore, the implementation of the additional inferential capabilities of Rational Closure is justified.

About future work, we have implemented Lexicographic Closure of the TBox and we plan to carry out an analogous evaluation (to the one presented in this paper) for it. We have also developed a semantics for Rational Closure of the ABox and a companion algorithm for computing this. We plan to implement this algorithm and to carry out a similar experimental evaluation of this. In addition, we plan to explore the feasibility of other rational consequence relations as candidates for defeasible entailment.

**Acknowledgements.** This work was partially funded by Project number 247601, Net2: Network for Enabling Networked Knowledge, from the FP7-PEOPLE-2009-IRSES call. It is also based upon research supported by the National Research Foundation.

## References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proc. of KR*, pages 673–684, 1992.
3. P. Bonatti, M. Faella, and L. Sauro. Defeasible inclusions in low-complexity DLs. *JAIR*, 42:719–764, 2011.
4. P. Bonatti, M. Faella, and L. Sauro. On the complexity of EL with defeasible inclusions. In *Proc. of IJCAI*, pages 762–767. AAAI Press, 2011.
5. P. Bonatti, C. Lutz, and F. Wolter. Description logics with circumscription. *Proc. of KR*, 6:400–41O, 2006.

6. P. Bonatti, C. Lutz, and F. Wolter. The complexity of circumscription in description logic. *JAIR*, 35(2):717, 2009.
7. K. Britz, G. Casini, T. Meyer, K. Moodley, and I. J. Varzinczak. Ordered Interpretations and Entailment for Defeasible Description Logics. Technical report, CAIR, CSIR Meraka and UKZN, South Africa, 2013.
8. K. Britz, T. Meyer, and I. Varzinczak. Semantic foundation for preferential description logics. In *Proc. of the Australasian Joint Conference on Artificial Intelligence*, number 7106 in LNAI, pages 491–500. Springer, 2011.
9. G. Casini and U. Straccia. Rational closure for defeasible description logics. In *Proc. of JELIA*, pages 77–90, 2010.
10. G. Casini and U. Straccia. Lexicographic closure for defeasible description logics. In *Proc. of Australasian Ontology Workshop*, volume 969, 2012.
11. F. M. Donini, D. Nardi, and R. Rosati. Autoepistemic description logics. In *Proc. of IJCAI*, volume 15, pages 136–141, 1997.
12. F. M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Transactions on Computational Logic (TOCL)*, 3(2):177–225, 2002.
13. C. Eardley, M. Kuhlmann, and A. Pauly. *The bee genera and subgenera of sub-Saharan Africa*. Belgian Development Cooperation, 2010.
14. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Preferential description logics. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 257–272. Springer, 2007.
15. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. A non-monotonic description logic for reasoning about typicality. *Artif. Intell.*, 195:165–202, 2013.
16. L. Giordano, N. Olivetti, V. Gliozzi, and G.L. Pozzato. $\mathcal{ALC} + T$: A preferential extension of description logics. *Fund. Inf.*, 96(3):341–372, 2009.
17. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amount: extracting modules from ontologies. In *Proceedings of the 16th international conference on World Wide Web*, pages 717–726. ACM, 2007.
18. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31(1):273–318, 2008.
19. J. Heinsohn. Probabilistic description logics. In *Proc. of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pages 311–318. Morgan Kaufmann Publishers Inc., 1994.
20. P. Ke and U. Sattler. Next steps for description logics of minimal knowledge and negation as failure. In *Proc. of the 2008 Description Logic Workshop (DL 2008)*, volume 353. Citeseer, 2008.
21. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Art. Intell.*, 44:167–207, 1990.
22. D. Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15:61–82, 1995.
23. D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Art. Intell.*, 55(1):1–60, 1992.
24. T. Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6):852–883, 2008.
25. K. Moodley, T. Meyer, and I. J. Varzinczak. A defeasible reasoning approach for description logic ontologies. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pages 69–78. ACM, 2012.
26. J. Pearl. System Z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proceedings of the 3rd conference on Theoretical aspects of reasoning about knowledge*, TARK '90, pages 121–135. Morgan Kaufmann Publishers Inc., 1990.

27. J. Quantz and M. Ryan. Preferential default description logics, 1993.
28. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial intelligence*, 48(1):1–26, 1991.
29. K. Sengupta, A. Krisnadhi, and P. Hitzler. Local closed world semantics: Grounded circumscription for OWL. In *Proc. of ISWC*, pages 617–632. Springer, 2011.
30. R. Stevens, M. E. Aranguren, K. Wolstencroft, U. Sattler, N. Drummond, M. Horridge, and A. Rector. Using OWL to model biological knowledge. *Intern. Journ. of Human-Computer Studies*, 65(7):583 – 594, 2007.