RAT-OWL: Reasoning with rational closure in description logics of typicality

Laura Giordano[1], Valentina Gliozzi[2], Gian Luca Pozzato[2], and Riccardo Renzulli[2]

[1] DiSIT, University of Piemonte Orientale "Amedeo Avogadro" - Italy - laura.giordano@uniupo.it

[2] Dipartimento di Informatica, Università di Torino, Italy - {gliozzi,pozzato}@di.unito.it, riccardo.renzulli@edu.unito.it

Abstract. We present RAT-OWL, a software system for reasoning about typicality in preferential Description Logics. It is implemented in the form of a Protégé 4.3 Plugin and it allows the user to reason in a nonmonotonic extension of Description Logics based on the notion of "rational closure". This logic extends standard Description Logics in order to express "typical" properties, that can be directly specified by means of a typicality operator $\mathbf{T}$: a TBox can contain inclusions of the form $\mathbf{T}(C) \sqsubseteq D$ to represent that "typical Cs are also Ds". We show experimental results, indicating that the performances of RAT-OWL are promising.

1 Introduction

Description Logics (DLs for short) are an important formalism of knowledge representation. DLs are reminiscent of the early semantic networks and of frame-based systems. They have been successfully implemented by many systems, and are at the basis of languages for the Semantic Web such as OWL. In a DL framework, a knowledge base (KB) comprises two components: a TBox, containing the definition of concepts (and possibly roles) and a specification of inclusion relations among them, and an ABox, containing instances of concepts and roles, in other words, properties and relations of individuals. In Description Logics one can use concept inclusion in order to express that all the members of a class have a given property (thus $Cat \sqsubseteq Mammal$ expresses the general property that "cats are mammals", and $Pet \sqsubseteq \exists hasOwner.\top$ that "all pets have an owner" (i.e. they are in the relation "hasOwner" with somebody). One can also use assertions in order to represent the fact that an individual has a given property, e.g. $Cat(tom)$ ("Tom is a cat") or $\exists hasOwner.\top(tom)$ ("Tom has an owner") or $hasOwner(tom, andrea)$ ("Andrea is Tom's owner"). A distinguishing quality of Description Logics is their controlled complexity: the trade-off between expressivity of the languages and good computational complexities is one of the main reasons justifying the success of DLs.

Since the very objective of a TBox is to formalize a taxonomy of concepts, the needs of representing properties holding only for *typical* individuals of a given concept (and not for *all* the elements), and thus to reason about defeasible inheritance in the presence of exceptions easily arise. In order to satisfy these needs, nonmonotonic extensions of DLs have been investigated since the early 90s [4, 1, 3, 12, 11, 25, 2]. A simple but powerful nonmonotonic extension of DLs is proposed in [14, 15, 17]: in this approach "typical" or "normal" properties can be directly specified by means of a "typicality" operator $\mathbf{T}$ enriching the underlying DL. The semantics of $\mathbf{T}$ is characterized by the core properties of nonmonotonic reasoning axiomatized by either *preferential logic* [20] or *rational logic* [22]. We focus on the Description Logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ introduced in [17]. In this logic one can express "defeasible inclusions" such as "normally, athletes are not fat people", $\mathbf{T}(\mathit{Athlete}) \sqsubseteq \neg \mathit{Fat}$, which do not hold for all individuals, and may have exceptions for some of them (e.g. for athletes which are sumo wrestlers). For instance, a knowledge base can consistently express that "normally, athletes are not fat people", whereas "sumo wrestlers are athletes that are typically fat" as follows:

$\mathit{SumoWrestler} \sqsubseteq \mathit{Athlete}$
$\mathbf{T}(\mathit{Athlete}) \sqsubseteq \neg \mathit{Fat}$
$\mathbf{T}(\mathit{SumoWrestler}) \sqsubseteq \mathit{Fat}$

In the Description Logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ standard models are extended by a function f which selects the typical/most normal instances of any concept C, i.e. the extension of $\mathbf{T}(C)$ is defined as $(\mathbf{T}(C))^{\mathcal{I}} = f(C^{\mathcal{I}})$. The function f satisfies a set of postulates that are a restatement of Kraus, Lehmann and Magidor's axioms of rational logic $\mathbf{R}$. This allows the typicality operator to inherit well-established properties of nonmonotonic entailment: for instance, if in addition "typical sumo wrestlers are strong" $\mathbf{T}(\mathit{SumoWrestler}) \sqsubseteq \mathit{Strong}$, we can conclude that "typical strong sumo wrestlers are fat" $\mathbf{T}(\mathit{SumoWrestler} \sqcap \mathit{Strong}) \sqsubseteq \mathit{Fat}$. It turns out that the semantics based on the selection function is equivalent to a semantics based on a *preference relation* $<$ among domain elements: intuitively, $x < y$ means that element x is more "normal" with respect to y, which is in some sense "exceptional" (such as a sumo wrestler which is not fat).

The logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ itself is too weak in several application domains. Indeed, although the operator $\mathbf{T}$ is nonmonotonic ($\mathbf{T}(C) \sqsubseteq E$ does not imply $\mathbf{T}(C \sqcap D) \sqsubseteq E$), the logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is monotonic, in the sense that if the fact F follows from a given knowledge base KB, then F also follows from any KB' $\supseteq$ KB. Furthermore, the inclusion $\mathbf{T}(\mathit{SumoWrestler} \sqcap \mathit{Blond}) \sqsubseteq \mathit{Fat}$ cannot be concluded, although being blond is *irrelevant* for a sumo wrestler, and we would like to conclude that a blond sumo wrestler is fat in absence of contrary evidence. In order to overcome this limitation and perform useful inferences, in [17] a nonmonotonic extension of the logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ has been introduced, based on a minimal model semantics, by extending to $\mathcal{ALC}$ the notion of *rational closure* defined by Lehmann and Magidor in [22] for propositional logic. The resulting logic, called $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$, supports nonmonotonic inference so that, in the example above, we are able to conclude that "typical strong sumo wrestlers are fat". The rational closure construction introduced in [17] retains the same complexity of the underlying description logic and, for $\mathcal{ALC}$, the problem of deciding whether a typicality inclusion $\mathbf{T}(C) \sqsubseteq D$ belongs to the rational closure of the TBox is in EXPTIME.

In this work we introduce RAT-OWL, a Protégé 4.3 Plugin for reasoning about typicality in the logic $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$. To the best of our knowledge, this is the first reasoner for nonmonotonic Description Logics of typicality with rational closure. RAT-OWL relies on a polynomial encoding of $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ in standard $\mathcal{ALC}$ introduced in [16], based on the definition of the typicality operator $\mathbf{T}$ in terms of a Gödel-Löb modality $\Box$, where $\mathbf{T}(C)$ is defined as $C \sqcap \Box \neg C$ and the accessibility relation of the modality $\Box$ corresponds to the preference relation $<$ in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ models. This allows us to rely on existing reasoners for standard DLs for constructing the rational closure.

The plan of the paper is as follows: in Section 2 we recall the definition of the description logic of typicality $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$, its encoding into $\mathcal{ALC}$ as well as the definition of the rational closure. In Section 3 we describe the implementation of RAT-OWL, and in Section 4 we present some experimental results witnessing its promising performance. In Section 5 we conclude by mentioning some related works and highlighting some pointers to future issues.

2 Preferential description logics

2.1 The monotonic logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$

The logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is obtained by adding to standard $\mathcal{ALC}$ the typicality operator $\mathbf{T}$ [14]. The intuitive idea is that $\mathbf{T}(C)$ selects the *typical* instances of a concept C. We can therefore distinguish between the properties that hold for all instances of concept C ($C \sqsubseteq D$), and those that only hold for the normal or typical instances of C ($\mathbf{T}(C) \sqsubseteq D$).

Definition 1. *We consider an alphabet of concept names $\mathcal{C}$, of role names $\mathcal{R}$, and of individual constants $\mathcal{O}$. Given $A \in \mathcal{C}$ and $R \in \mathcal{R}$, we define:*

$$C_R := A \mid \top \mid \bot \mid \neg C_R \mid C_R \sqcap C_R \mid C_R \sqcup C_R \mid \forall R.C_R \mid \exists R.C_R$$
$$C_L := C_R \mid \mathbf{T}(C_R)$$

A knowledge base is a pair $(\mathcal{T}, \mathcal{A})$. $\mathcal{T}$ contains a finite set of concept inclusions $C_L \sqsubseteq C_R$. $\mathcal{A}$ contains assertions of the form $C_L(a)$ and $R(a, b)$, where $a, b \in \mathcal{O}$.

The semantics of the $\mathbf{T}$ operator can be given by means of a set of postulates that are a reformulation of axioms and rules of nonmonotonic entailment in rational logic **R** [22]: in this respect an assertion of the form $\mathbf{T}(C) \sqsubseteq D$ is equivalent to the conditional assertion $C \mid\sim D$ in **R**. The basic idea is to extend the notion of $\mathcal{ALC}$ interpretation with a selection function. Let $\langle \Delta^{\mathcal{I}}, .^{\mathcal{I}} \rangle$ be an $\mathcal{ALC}$ interpretation, where Δ is a set of elements (the domain); $.^{\mathcal{I}}$ is an extension function that maps each concept C to $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each role R to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each individual constant $a \in \mathcal{O}$ to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; and, for all concepts of $\mathcal{ALC}$, C^I is defined in the usual way. We define function $f_{\mathbf{T}} : Pow(\Delta^{\mathcal{I}}) \longmapsto Pow(\Delta^{\mathcal{I}})$ that selects the *typical* instances of any $S \subseteq \Delta^{\mathcal{I}}$; for $S = C^{\mathcal{I}}$, the selection function selects the typical instances of concept C, namely: $(\mathbf{T}(C))^{\mathcal{I}} = f_{\mathbf{T}}(C^{\mathcal{I}})$. Function $f_{\mathbf{T}}$ has the following properties for all $S \subseteq \Delta^{\mathcal{I}}$, that are essentially a restatement of the properties characterizing rational entailment:

$(f_{\mathbf{T}} - 1)$ $f_{\mathbf{T}}(S) \subseteq S$
$(f_{\mathbf{T}} - 2)$ if $S \neq \emptyset$, then also $f_{\mathbf{T}}(S) \neq \emptyset$
$(f_{\mathbf{T}} - 3)$ if $f_{\mathbf{T}}(S) \subseteq R$, then $f_{\mathbf{T}}(S) = f_{\mathbf{T}}(S \cap R)$
$(f_{\mathbf{T}} - 4)$ $f_{\mathbf{T}}(\bigcup S_i) \subseteq \bigcup f_{\mathbf{T}}(S_i)$
$(f_{\mathbf{T}} - 5)$ $\bigcap f_{\mathbf{T}}(S_i) \subseteq f_{\mathbf{T}}(\bigcup S_i)$
$(f_{\mathbf{T}} - 6)$ if $f_{\mathbf{T}}(S) \cap R \neq \emptyset$, then $f_{\mathbf{T}}(S \cap R) \subseteq f_{\mathbf{T}}(S)$

Observe these properties are stronger with respect to the properties of “normality concepts” in [6], which do not need to satisfy all the postulates. The semantics of $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ can equivalently be formulated in terms of rational models: ordinary models of $\mathcal{ALC}$ are equipped with a $preference\ relation$ $<$ on the domain, whose intuitive meaning is to compare the “typicality” of domain elements, that is to say, $x < y$ means that x is more typical than y. Typical members of a concept C, that is members of $\mathbf{T}(C)$, are the members x of C that are minimal with respect to this preference relation (such that there is no other member of C more typical than x).

Definition 2 (Semantics of $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$). *A model $\mathcal{M}$ of $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is any structure $\langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$ where: Δ and $.^{\mathcal{I}}$ are defined as in $\mathcal{ALC}$ interpretations and $<$ is an irreflexive, transitive and modular (if $x < y$ then either $x < z$ or $z < y$) binary relation over $\Delta^{\mathcal{I}}$; For the $\mathbf{T}$ operator, we let $(\mathbf{T}(C))^{\mathcal{I}} = Min_<(C^{\mathcal{I}})$, where $Min_<(S) = \{u : u \in S$ and $\nexists z \in S$ s.t. $z < u\}$. Also, $<$ satisfies the $Well\text{-}Foundedness\ Condition$, i.e., for all $S \subseteq \Delta^{\mathcal{I}}$, for all $x \in S$, either $x \in Min_<(S)$ or $\exists y \in Min_<(S)$ such that $y < x$.*

Definition 3 (Model satisfying a knowledge base). *Given a model $\mathcal{M}= \langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$, we say that: (i) a model $\mathcal{M}$ satisfies an inclusion $C \sqsubseteq D$ (written $\mathcal{M} \models_{\mathcal{ALC}^{\mathbf{R}}_{RaCl}\mathbf{T}} C \sqsubseteq D$) if it holds $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; (ii) $\mathcal{M}$ satisfies an assertion $C(a)$ (written $\mathcal{M} \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} C(a)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$; (iii) $\mathcal{M}$ satisfies an assertion $R(a, b)$ (written $\mathcal{M} \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} R(a, b)$) if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. Given a knowledge base K=$(\mathcal{T}, \mathcal{A})$, we say that: $\mathcal{M}$ satisfies $\mathcal{T}$ if $\mathcal{M}$ satisfies all inclusions in $\mathcal{T}$; $\mathcal{M}$ satisfies $\mathcal{A}$ if $\mathcal{M}$ satisfies all assertions in $\mathcal{A}$; $\mathcal{M}$ satisfies K if it satisfies both $\mathcal{T}$ and $\mathcal{A}$; a concept C is satisfiable with respect to K, if there is $\mathcal{M} = \langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$ satisfying K and such that $C^{\mathcal{I}} \neq \emptyset$.*

Let us define entailment of an inclusion in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$:

Definition 4. *Given a knowledge base K, an inclusion $C_L \sqsubseteq C_R$ is* entailed *from K, written $K \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} C_L \sqsubseteq C_R$, if ${C_L}^{\mathcal{I}} \subseteq {C_R}^{\mathcal{I}}$ holds in all models $\mathcal{M} = \langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$ satisfying K.*

Definition 5 (Rank of a domain element $k_{\mathcal{M}}(x)$). *Given a model $\mathcal{M} = \langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$, the rank $k_{\mathcal{M}}$ of a domain element $x \in \Delta^{\mathcal{I}}$, is the length of the longest chain $x_0 < \cdots < x$ from x to a minimal x_0 (i.e. such that there is no x' such that $x' < x_0$).*

The rank function $k_{\mathcal{M}}$ and $<$ can be defined from each other by letting $x < y$ if and only if $k_{\mathcal{M}}(x) < k_{\mathcal{M}}(y)$.

Definition 6 (Rank of a concept $k_{\mathcal{M}}(C_R)$ in $\mathcal{M}$). *Given a model $\mathcal{M} = \langle \Delta^{\mathcal{I}}, <, .^{\mathcal{I}} \rangle$, the* rank $k_{\mathcal{M}}(C_R)$ of a concept C_R in the model $\mathcal{M}$ *is defined as $k_{\mathcal{M}}(C_R) = min\{k_{\mathcal{M}}(x) \mid x \in {C_R}^{\mathcal{I}}\}$. If ${C_R}^{\mathcal{I}} = \emptyset$, then C_R has no rank and we write $k_{\mathcal{M}}(C_R) = \infty$.*

It is immediate to see that, for any $\mathcal{M}$, we have that $\mathcal{M}$ satisfies $\mathbf{T}(C) \sqsubseteq D$ if and only if $k_{\mathcal{M}}(C \sqcap D) < k_{\mathcal{M}}(C \sqcap \neg D)$.

2.2 Reasoning in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$

In order to reason in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$, we exploit the following polynomial encoding of a KB in standard $\mathcal{ALC}$. The encoding was originally defined in [18] for the more expressive logic $\mathcal{SHIQ}$, but it works for $\mathcal{ALC}$ as well. The idea is to exploit a definition of the typicality operator $\mathbf{T}$ in terms of a Gödel-Löb modality $\Box$ by defining $\mathbf{T}(C)$ as $C \sqcap \Box\neg C$, where the accessibility relation of the modality $\Box$ is the preference relation $<$ in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ models ($\Box_{\neg C}$ meaning that more preferred domain elements are instances of $\neg C$).

Let KB=($\mathcal{T}, \mathcal{A}$) be a knowledge base where $\mathcal{A}$ does not contain positive typicality assertions on individuals of the form $\mathbf{T}(C)(a)$. We define the encoding KB'=($\mathcal{T}', \mathcal{A}'$) of KB in $\mathcal{ALC}$ as follows. First of all, we let $\mathcal{A}' = \mathcal{A}$. Then, for each $C \sqsubseteq D \in \mathcal{T}$, not containing $\mathbf{T}$, we introduce $C \sqsubseteq D$ in $\mathcal{T}'$. For each $\mathbf{T}(C)$ occurring in $\mathcal{T}$, we introduce a new atomic concept $\Box_{\neg C}$ and, for each inclusion $\mathbf{T}(C) \sqsubseteq_n D \in \mathcal{T}$, we add to $\mathcal{T}'$ the inclusion $C \sqcap \Box_{\neg C} \sqsubseteq D$. To capture the properties of $\Box$ modality, a new role R is introduced to represent the relation $<$ and the following inclusions are introduced in $\mathcal{T}'$:

$$\Box_{\neg C} \sqsubseteq \forall R.(\neg C \sqcap \Box_{\neg C}) \qquad\qquad \neg\Box_{\neg C} \sqsubseteq \exists R.(C \sqcap \Box_{\neg C})$$

The first inclusion accounts for the transitivity of $<$. The second inclusion accounts for the well-foundedness, namely the fact that if an element is not a typical C element then there must be a typical C element preferred to it. For the encoding of a query, $C_L \sqsubseteq C_R$, if $C_L \sqsubseteq C_R$ is not a typicality inclusion, then $C'_L = C_L$ and $C'_R = C_R$; if $C_L \sqsubseteq C_R$ is a typicality inclusion $\mathbf{T}(C) \sqsubseteq C_R$, then $C'_L = C \sqcap \Box_{\neg C}$ and $C'_R = C_R$.

The size of KB' is polynomial in the size of the KB. The same for the encoding of a query $C_L \sqsubseteq C_R$, assuming its size is polynomial in the size of K.

Given the above encoding, in [16] it is shown that:

$$\text{KB} \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} C_L \sqsubseteq C_R \text{ if and only if KB' } \models_{\mathcal{ALC}} C'_L \sqsubseteq C'_R$$

and, as a consequence, that the problem of deciding entailment in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is in EXPTIME, since reasoning in $\mathcal{ALC}$ is in EXPTIME. EXPTIME-hardness follows from the fact that entailment in $\mathcal{ALC}$ is EXPTIME-hard, so that the problem of deciding entailment in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is EXPTIME-complete.

2.3 The nonmonotonic logic $\mathcal{ALC}^{\mathbf{R}}_{\mathbf{RaCl}}\mathbf{T}$

As mentioned above, although the typicality operator $\mathbf{T}$ itself is nonmonotonic, the logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is monotonic: what is inferred from K can still be inferred from any K' with $K \subseteq K'$. This is a clear limitation in DLs. As a consequence of the monotonicity of $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$, one cannot deal with irrelevance, for instance. So one cannot derive from $K = \{\mathit{SumoWrestler} \sqsubseteq \mathit{Athlete}, \mathbf{T}(\mathit{Athlete}) \sqsubseteq \neg\mathit{Fat}, \mathbf{T}(\mathit{SumoWrestler}) \sqsubseteq \mathit{Fat}\}$ that $K \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} \mathbf{T}(\mathit{SumoWrestler} \sqcap \mathit{Bald}) \sqsubseteq \mathit{Fat}$, even if the property of being bald is irrelevant with respect to being fat or not. In order to perform useful nonmonotonic inferences, in [17] the authors have strengthened the above semantics by restricting entailment to a class of minimal models. Intuitively, the idea is to restrict entailment to minimal models which *minimize the atypicality of concepts*. For ease of notation, we call the resulting logic $\mathcal{ALC}^{\mathbf{R}}_{\mathrm{RaCl}}\mathbf{T}$, as minimally entailed inclusions are those belonging to the *rational closure* of the KB [17]. In the following we recall the definition of the rational closure for $\mathcal{ALC}$, which is a natural extension of the rational closure in [22]. We first recall the notion of *exceptionality* of concepts and inclusions w.r.t. a KB.

Definition 7 (Exceptionality of concepts and inclusions). *Let K=($\mathcal{T}, \mathcal{A}$) be a knowledge base. A concept C is said to be* exceptional *for K if and only if $K \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} \mathbf{T}(\top) \sqsubseteq \neg C$. A $\mathbf{T}$-inclusion $\mathbf{T}(C) \sqsubseteq D$ is exceptional for K if C is exceptional for K. The set of $\mathbf{T}$-inclusions of K which are exceptional in K will be denoted as $\mathcal{E}(K)$.*

Definition 8. *Given a knowledge base* K=$(\mathcal{T}, \mathcal{A})$*, it is possible to define a sequence of knowledge bases* $E_0, \ldots, E_i, \ldots, E_n$ *by letting* $E_0 = (\mathcal{T}_0, \mathcal{A})$ *where* $\mathcal{T}_0 = \mathcal{T}$ *and for* $i > 0$*,* $E_i = (\mathcal{T}_i, \mathcal{A})$ *where* $\mathcal{T}_i = \mathcal{E}(E_{i-1}) \cup \{C \sqsubseteq D \in \mathcal{T} \mid \mathbf{T}$ *does not occur in* $C\}$*.*

Clearly $\mathcal{T}_0 \supseteq \mathcal{T}_1 \supseteq \mathcal{T}_2, \ldots$. Observe that, being K finite, there is a least $n \geq 0$ such that, for all $m > n, \mathcal{T}_m = \mathcal{T}_n$ or $\mathcal{T}_m = \emptyset$. We take $(\mathcal{T}_n, \mathcal{A})$ as the last element of the sequence of knowledge bases starting from K.

Definition 9 (Rank of a concept). *A concept* C *has* rank i *(denoted by* $rank(C) = i$*) for* K=$(\mathcal{T}, \mathcal{A})$*, if and only if* i *is the least natural number for which* C *is not exceptional for* E_i*. If* C *is exceptional for all* E_i *then* $rank(C) = \infty$*, and we say that* C *has no rank.*

From the above definition it follows that if a concept C has a rank, its highest possible value is n. The notion of rank of a formula allows one to define the rational closure of a knowledge base K with respect to TBox .

Definition 10 (Rational closure of TBox). *Let* K=$(\mathcal{T}, \mathcal{A})$ *be a knowledge base. We define* $\overline{\mathcal{T}}$*, the* rational closure *of* $\mathcal{T}$*, as* $\overline{\mathcal{T}} = \{\mathbf{T}(C) \sqsubseteq D \mid$ *either* $rank(C) < rank(C \sqcap \neg D)$ *or* $rank(C) = \infty\} \cup \{C \sqsubseteq D \mid K \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} C \sqsubseteq D\}$*.*

The nonmonotonic semantics of $\mathcal{ALC}^{\mathbf{R}}_{\mathrm{RaCl}}\mathbf{T}$ relies on minimal rational models that minimize the *rank of domain elements*. Informally, given two models of a KB, one in which a given domain element x has rank 2 (because for instance $z < y < x$), and another in which it has rank 1 (because only $y < x$), we prefer the latter, as in this model the element x is assumed to be "more typical" than in the former. More precisely, we have that $\mathcal{M} < \mathcal{M}'$ if, for all $x \in \Delta^{\mathcal{I}}$, it holds that $k_{\mathcal{M}}(x) \leq k_{\mathcal{M}'}(x)$ whereas there exists $y \in \Delta^{\mathcal{I}}$ such that $k_{\mathcal{M}}(y) < k_{\mathcal{M}'}(y)$. Given a knowledge base K, we say that $\mathcal{M}$ is a minimal model of K with respect to $<$ if it is a model satisfying K and there is no $\mathcal{M}'$ model satisfying K such that $\mathcal{M}' < \mathcal{M}$. For technical reasons, we further need to restrict our attention to *canonical models* (we refer to [17] for the definition). Query entailment in $\mathcal{ALC}^{\mathbf{R}}_{\mathrm{RaCl}}\mathbf{T}$ is then restricted to minimal canonical models: an inclusion $C_L \sqsubseteq C_R$ is entailed from K in $\mathcal{ALC}^{\mathbf{R}}_{\mathrm{RaCl}}\mathbf{T}$, written $K \models_{\mathcal{ALC}^{\mathbf{R}}_{\mathrm{RaCl}}\mathbf{T}} C_L \sqsubseteq C_R$, if $C_L \sqsubseteq C_R$ holds in all minimal canonical models of K with respect to $\mathcal{T}$. In [17] it is shown that minimal entailment provides a semantic characterization of rational closure, so that a query $C_L \sqsubseteq C_R$ is in $\overline{\mathcal{T}}$ if and only if $K \models_{\mathcal{ALC}^{\mathbf{R}}_{\mathrm{RaCl}}\mathbf{T}} C_L \sqsubseteq C_R$, and that the problem of deciding whether $\mathbf{T}(C) \sqsubseteq D \in \overline{\mathcal{T}}$ is in EXPTIME, the same complexity upper bound of entailment in the underlying description logic $\mathcal{ALC}$ [13].

3 Design of RAT-OWL

As mentioned in the Introduction, since the main aim of Description Logics is to build taxonomies, the need of representing inheritance with exceptions, prototypical and defeasible properties easily arises. Protégé does not allow users to reason directly about these features. Thus, RAT-OWL, which stands for **RAtional closure with Typicality in OWL**, is intended to meet these needs.

RAT-OWL[3] is a Protégé 4.3 Plugin and it is written in Java and heavily uses OWL API 3.4 to manipulate OWL ontologies. It is based on the $\mathcal{ALC}^{\mathbf{R}}_{\mathrm{RaCl}}\mathbf{T}$ logic, in detail

[3] https://drive.google.com/folderview?id=0BzebarfrIf_kc3RqcmR4T1BwVzg

$\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ extended with the notion of rational closure of the TBox in order to perform nonmonotonic inferences. RAT-OWL makes use of the polynomial encoding into $\mathcal{ALC}$ described in Section 2.2. As an example, let the TBox contain:

1. $\mathbf{T}(Bird) \sqsubseteq Fly$
2. $\mathbf{T}(Penguin) \sqsubseteq \neg Fly$
3. $Penguin \sqsubseteq Bird$

so, the encoding KB'[4] contains:

1. $Bird \sqcap Bird1 \sqsubseteq Fly$
   $Bird1 \sqsubseteq \forall R.(\neg Bird \sqcap Bird1)$
   $\neg Bird1 \sqsubseteq \exists R.(Bird \sqcap Bird1)$
2. $Penguin \sqcap Penguin1 \sqsubseteq \neg Fly$
   $Penguin1 \sqsubseteq \forall R.(\neg Penguin \sqcap Penguin1)$
   $\neg Penguin1 \sqsubseteq \exists R.(Penguin \sqcap Penguin1)$
3. $Penguin \sqsubseteq Bird$

and in Manchester OWL syntax[5]:

1. $Bird$ **and** $Bird1$ **SubClassOf** Fly
   $Bird1$ **SubClassOf** $(R$ **only** $($**not** $Bird$ **and** $Bird1))$
   not $Bird1$ **SubClassOf** $(R$ **some** $(Bird$ **and** $Bird1))$
2. $Penguin$ **and** $Penguin1$ **SubClassOf not** Fly
   $Penguin1$ **SubClassOf** $(R$ **only** $($**not** $Penguin$ **and** $Penguin1))$
   not $Penguin1$ **SubClassOf** $(R$ **some** $(Penguin$ **and** $Penguin1))$
3. $Penguin$ **SubClassOf** $Bird$

In order to reason about typicality in Protégé, one could in principle manually do the above encoding. However, RAT-OWL does the same encoding in an automatic way. Once a class has been added in the active ontology, it is possible to add the corresponding typical class just by selecting the class and then clicking on the **T** icon, next to the sibling icon, in the **Typical Class Hierarchy View** on the left hand side. As a result, the typical class is created and the encoding is done automatically. For instance, if one wants to reason about typical birds, the following axioms are added to the ontology:

1. $\mathbf{T}(Bird)$ **EquivalentTo** $(Bird$ **and** $Bird1)$
2. $NotBird1$ **EquivalentTo** $($**not** $(Bird1))$
3. $Bird1$ **SubClassOf** $(R$ **only** $($**not** $Bird$ **and** $Bird1))$
4. $NotBird1$ **SubClassOf** $(R$ **some** $(\mathbf{T}(Bird)))$
5. $\mathbf{T}(Bird)$ **comment** *A typical class for reasoning about typicality*
6. $Bird1$ **comment** *An auxiliary class for reasoning about typicality@en*
7. $NotBird1$ **comment** *An auxiliary class for reasoning about typicality@en*

Notice that, given a class *A*, classes *NotA*1 and *A*1 are added to the ontology too. In order to improve the readability of the resulting hierarchy, the "auxiliary" classes *NotA*1 and *A*1 are hidden in the hierarchy. This is implemented by means of OWLAnnotations.

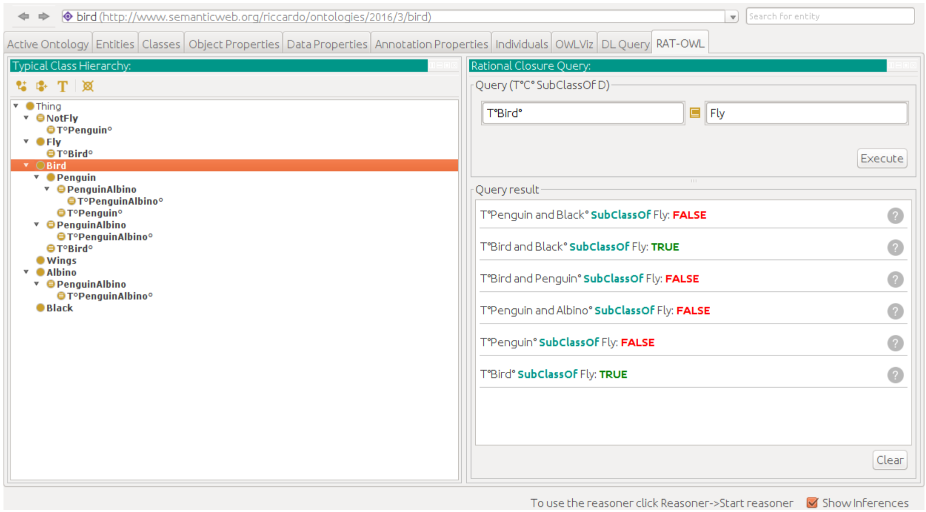

Fig. 1. RAT-OWL tab

Figure 1 illustrates the plugin interface. On the left hand side of the window there is the hierarchy. On the right hand side there is the **Rational Closure Query View** where the user can write both classical queries such as $C \sqsubseteq D$ and typicality queries such as $\mathbf{T}(C) \sqsubseteq D$. In the former case, calling directly the reasoner selected from the Protégé user interface is enough whereas, in the latter case, first rational closure construction is needed, then $rank(C)$ and $rank(C \sqcap \neg D)$ are computed; as illustred in Definition 10, $\mathbf{T}(C) \sqsubseteq D$ is entailed by $\overline{\mathcal{T}}$ if and only if $rank(C) < rank(C \sqcap \neg D)$. The rational closure of the TBox of the active ontology is computed once and for all when the first query is considered. If the knowledge base does not change, the same construction is kept in order to answer subsequent queries.

RAT-OWL is accessible through the Protégé user interface in the *Window* menu and it can be used as any other Protégé plugin. Below is the abstract description of the algorithm *RationalClosureSetup* for computing the ranking of the knowledge base.

From an implementation point of view, in order to save memory space during the computation of rational closure levels, as can be seen in Listing 1.1, first *commonOntology* is computed, i.e. the ontology including the strict inclusions which are common to all the levels; then, for each level, only the set of exceptional concepts is recorded, rather then the set of rules in E_i.

The cycle in Listing 1.1 computes rational closure levels and, for each level, the method *exceptionalConceptsAndInclusions* is called in which concept ranks are updated in *rankMap*, exceptional axioms are saved as OWLOntology and finally they are added

[4] In this implementation auxiliary concepts of a concept C are called $C1$ instead of $\Box\neg C$.

[5] Manchester OWL syntax is a user-friendly syntax for OWL DL that is fundamentally based on collecting all information about a particular class, property, or individual into a single construct, called a frame.

Algorithm 1 RationalClosureSetup($\mathcal{T}$)

Input: A TBox $\mathcal{T}$
Output: A list of sets of exceptional concepts $\mathcal{C}_0$,$\mathcal{C}_1$,...,$\mathcal{C}_n$
$\mathcal{S} \leftarrow \{C \sqsubseteq D \in \mathcal{T} \mid \mathbf{T}$ does not occur in $C\}$ ▷ Let $\mathcal{S}$ be the set of strict inclusions in $\mathcal{T}$
$\mathcal{W} \leftarrow \mathcal{T}^{\square} \setminus \mathcal{S}$ ▷ $\mathcal{T}^{\square}$ is the $\mathcal{ALC}$ encoding of $\mathcal{T}$
$\mathcal{C} \leftarrow \{C \mid \mathbf{T}(C) \sqsubseteq D \in \mathcal{W}\}$
$\mathcal{C}_0 \leftarrow$ ExceptionalConcepts($\mathcal{W}$, $\mathcal{S}$, $\mathcal{C}$)
$\mathcal{E}_0 \leftarrow \{\mathbf{T}(C) \sqsubseteq D \in \mathcal{W} \mid C \in \mathcal{C}_0\}$
$i \leftarrow 0$
repeat
    $\mathcal{C}_{i+1} \leftarrow$ ExceptionalConcepts($\mathcal{E}_i$, $\mathcal{S}$, $\mathcal{C}_i$)
    $\mathcal{E}_{i+1} \leftarrow \{\mathbf{T}(C) \sqsubseteq D \in \mathcal{W} \mid C \in \mathcal{C}_{i+1}\}$
    $i \leftarrow i + 1$
until ($\mathcal{C}_i = \emptyset$ or $\mathcal{C}_i = \mathcal{C}_{i-1}$)
if $\mathcal{C}_i = \mathcal{C}_{i-1}$ **then**
    return ($\mathcal{C}_0$,$\mathcal{C}_1$,...,$\mathcal{C}_{i-1}$)
else
    return ($\mathcal{C}_0$,$\mathcal{C}_1$,...,$\mathcal{C}_i$)

Algorithm 2 ExceptionalConcepts($\mathcal{E}$,$\mathcal{S}$,$\mathcal{C}$)

Input: Encoding of a set of typicality inclusions $\mathcal{E}$, a set of strict inclusions $\mathcal{S}$, a set of concepts $\mathcal{C}$
Output: The set of exceptional concepts w.r.t. $\mathcal{E} \cup \mathcal{S}$
$\mathcal{C}_1 \leftarrow \{C \in \mathcal{C} \mid \mathcal{E} \cup \mathcal{S} \models_{\mathcal{ALC}} \mathbf{T}(Thing) \sqsubseteq \neg C\}$
return $\mathcal{C}_1$

to *subsets*. Notice that in this implementation $\mathcal{E}(E_i)$ contains all $\mathbf{T}$-inclusion $\mathbf{T}(C) \sqsubseteq D$ such that C is exceptional for E_i in addition to $C1$ and $NotC1$ referring axioms. Furthermore, the rank of auxiliary concepts $C1$ and $NotC1$ are not calculated.

The reasoner used by default in our plugin is the one selected by the user in the Protégé user interface and its root ontology is exactly *commonOntology*. In order to take advantage of inferences already predicted by the reasoner, all levels will be imported in the root ontology every time it is needed.

```
public class RationalClosure {

  private ArrayList<OWLOntology> subsets;
  private HashMap<String,Integer> rankMap;
  private OWLReasonerFactory reasonerFactory;
  private OWLReasoner reasoner;
  private OWLOntology ontology;
  private OWLOntology commonOntology;
  private OWLOntologyManager manager;
  private OWLDataFactory dataFactory;
  private long timeOut;
  private boolean full;
  ...
  public void setup() throws OWLOntologyCreationException {
    long start = System.currentTimeMillis();
    int level = 0;
    Set<OWLAxiom> tAxioms = getTypicalAxioms();
    this.commonOntology = getCommonOntology(tAxioms);
```

```
    //Example: reasonerFactory = new FaCTPlusPlusReasonerFactory();
    this.reasoner = reasonerFactory.createNonBufferingReasoner(commonOntology);
    this.rankMap = initialiseRankMap();
    Set<OWLAxiom> e0 = exceptionalConceptsAndInclusions(tAxioms,level);
    Set<OWLAxiom> e0copy = null;
    Set<OWLAxiom> e1 = new HashSet<OWLAxiom>();

    if (!e0.isEmpty()) {
      subsets.add(manager.createOntology(e0));
        do {
          level++;
          e0copy = new HashSet<OWLAxiom>(e0);
          e1 = exceptionalConceptsAndInclusions(e0, level);
          e0 = new HashSet<OWLAxiom>(e1);
          e0copy.removeAll(e1);
          if (!e1.isEmpty() && !e0copy.isEmpty())
          subsets.add(manager.createOntology(e1));
        } while (!e0copy.isEmpty());
    }

    //set rank of concepts which are exceptionals in all levels
    for (Map.Entry<String, Integer> entry : rankMap.entrySet()) {
         if(entry.getValue() > level)
           entry.setValue(Integer.MAX_VALUE);
    }

    subsets.add(null);
    timeOut = System.currentTimeMillis() - start;
  }
...
}
```

Listing 1.1. RationalClosure.java

When the user writes a typicality query such as $\mathbf{T}(C) \sqsubseteq D$, as can be seen in Listing 1.2, if $rank(C)$ is already present in *rankMap*, it is not calculated again from level 0. For example this is the case when $\mathbf{T}(C)$ is already contained in the KB. Furthermore, it is evident that $rank(C \sqcap \neg D)$ is at least equal to $rank(C)$, and we do not need to start from level 0, but from level $rank(C)$ to check the exceptionality of $C \sqcap \neg D$.

On the other hand, if $rank(C)$ was not already calculated, then method *calculateRank* is called, in which *subsets.get(i)* is imported in the *commonOntology* in order to check if C is exceptional at level *i*, for all *i* $\in$ *[0, subsets.size() - 1]*. Notice that *OWLTypicalClass* is a class which *extends OWLClass* and it is introduced by us in order to simplify the manipulation of typical classes in OWL. We will see in the next section how rational closure can be built up in two different ways.

```
private RationalClosureQueryResult executeRationalClosureQuery(OWLSubClassOfAxiom
     axiom) {
  RationalClosureQueryResult result = new RationalClosureQueryResult();

  OWLClassExpression exprSubClass = axiom.getSubClass();
  OWLClassExpression exprSuperClass = axiom.getSuperClass();
  Integer rankLeft = null;

  OWLTypicalClass subClass = (OWLTypicalClass) exprSubClass.asOWLClass();
  OWLClassExpression innerExpr = subClass.getInnerClassExpression();

  if (!innerExpr.isAnonymous()) {
    rankLeft = rClosure.getRankMap().get(innerExpr.asOWLClass().toStringID());
    if (rankLeft == null)
      rankLeft = rClosure.calculateRank(innerExpr, 0);
  } else {
    rankLeft = rClosure.calculateRank(innerExpr, 0);
```

```
  }

  int rankRight = rClosure.calculateRank(rClosure.getOWLDataFactory().
     getOWLObjectIntersectionOf(innerExpr,
rClosure.getOWLDataFactory().getOWLObjectComplementOf(exprSuperClass)),rankLeft);

  result.setQuery(axiom.toString());
  result.setRankLeft(rankLeft);
  result.setRankRight(rankRight);
  result.setResult(rankLeft < rankRight);

  return result;
}
```

Listing 1.2. RationalClosureQuery.java

4 Performances of RAT-OWL

We tested rational closure construction and query entailment on some test suites kindly provided by Bonatti et al. [5]. These test suites were obtained by modifying a version of the Gene Ontology (GO) published in 2006, which contains 20465 atomic concepts and 28896 concept inclusions. Test suites differ in *CI-to-DI-rate* and *DA-rate* parameters. The former controls the percentage of strict inclusions transformed into defeasible ones (in our case $C \sqsubseteq D$ is transformed into $\mathbf{T}(C) \sqsubseteq D$), and hence the overall percentage of defeasible inclusions in the KB. The latter controls the percentage of random disjointness axioms injected in order to increase the number of conflicts between defeasible inclusions. The experiments were performed on an Intel i7-5500U CPU 2.4 GHz with 8 GB RAM and Ubuntu 16.04 LTS in Java 8 with the options *-Xms3G -Xmx6G*. For each parameter configuration we report the execution time of the rational closure construction and of query entailment. The reported values are obtained by averaging execution times over ten nonmonotonic ontologies and fifty queries on each ontology.

As reasoners HermiT 1.3.8 and Fact++ 1.6.3 were used. For each parameter configuration we report the average execution time of the rational closure contruction (Figure 2) and of query entailment (Figure 3). It can be seen in Figure 2 that HermiT is much slower than Fact++ in building up the rational closure so the second reasoner was preferred for the most part of the tests. On the other hand, HermiT is much faster than Fact++ in query entailment. A possible explanation, to be verified, is that most of the queries are subsumptions $\mathbf{T}(C) \sqsubseteq D$ where $\mathbf{T}(C)$ does not occur in the ontology and the rank of C needs to be computed from scratch. Furthermore, for each parameter setting, rational closure was built up in two modalities: *full* and *restricted*. With the first one, the *rank* of any concept in the ontology is computed, while, with the second one, only the *rank* of concepts C such that $\mathbf{T}(C)$ occurs in the ontology is computed. Figures 2 and 3 report the time needed with the full modality. We observe that, with the restricted modality, as expected, the rational closure construction time decreases (in Table 1, by a factor of 3 for *DA-rate* less that 25% and, in Table 2, by a factor of 3.5 for *CI-to-DI-rate* less then 15%), while query entailment time increases.

Thus, for practical application of the rational closure of the TBox, the restricted modality may be preferred when the ontology is large but the *CI-to-DI-rate* (percentage of defeasible inclusions) is low (below 15%).

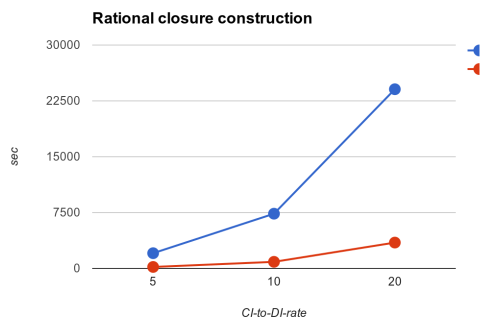

Fig. 2. HermiT and Fact++ rational closure construction time (15% DA-rate, full)

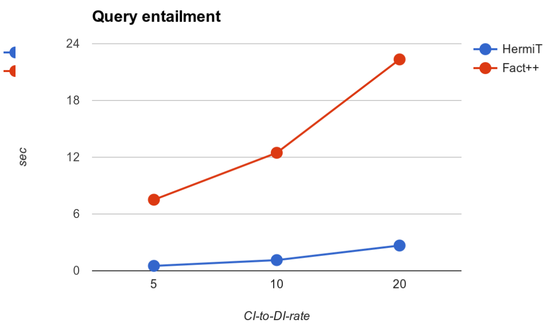

Fig. 3. HermiT and Fact++ query entailment time (15% DA-rate, full)

DA-rate	Rational closure	Query
05%	1642.60	8.87
10%	3134.58	12.73
20%	3378.90	22.38
25%	3112.81	22.89
30%	4213.91	23.63

DA-rate	Rational closure	Query
05%	445.87	18.89
10%	700.56	26.38
20%	905.48	39.57
25%	1046.19	51.09
30%	1226.47	62.24

Table 1. Fact++ rational closure construction and query time (sec) (15% *CI-to-DI-rate*, full and restricted)

CI-to-DI-rate	Rational closure	Query
05%	200.78	7.51
10%	887.92	12.47
15%	2520.91	17.42
20%	3473.62	22.34
25%	8316.73	39.88

CI-to-DI-rate	Rational closure	Query
05%	37.81	14.95
10%	191.00	26.25
15%	692.46	35.67
20%	2048.55	50.80
25%	6812.30	81.15

Table 2. Fact++ rational closure construction and query time (sec) (15% *DA-rate*, full and restricted)

Results in Table 2 also show how the increasing percentage of typicality inclusions negatively affects the execution time as expected. This is because for each typical class, as seen in section 3, seven new OWLAxiom are added to the ontology and so the higher *CI-to-DI-rate* is, the more expensive the class hierarchy step of an OWLReasoner is.

As a term of comparison we can take Bonatti et al. [5]'s *naive* method results and even though we did not use any modularization techniques, our experimental results are overall better and, thus, very promising. It has to be noted, however, that the approach in [5] allows for a more sophisticated treatment of inheritance and overriding with respect to rational closure, which does not allow an independent treatment of different defeasible properties of a concept. We refer to the conclusions for some discussion.

5 Conclusions and future works

We have presented RAT-OWL, a software system allowing a user to reason about typicality in Description Logics in an extension of standard DLs based on the well established nonmonotonic mechanism of rational closure. Experimentation over test suites developed in [5], which modifies a version of the Gene Ontology making a percentage of inclusions defeasible, witnesses that performance of RAT-OWL is promising.

The rational closure of a knowledge base has been introduced by Lehmann and Magidor [22] to allow for stronger inferences with respect to preferential and rational entailment, and several constructions of rational closure have been proposed for the description logic $\mathcal{ALC}$ [9, 8, 17]. All such constructions are defined for knowledge bases containing strict or defeasible inclusions, that in our approach are expressed as typicality inclusions. One major difference between our construction and those is [9, 8] is in the notion of exceptionality: our definition exploits preferential entailment, while [9, 8] directly use entailment in $\mathcal{ALC}$ over a materialization of the KB. In [24] a Defeasible-Inference Platform for OWL Ontologies has been proposed for the rational closure in [8] and in [23] a new algorithm for computing rational closure has been develped for $\mathcal{ALC}$, exploiting materialization of the KB and reasoning in $\mathcal{ALC}$ with a Protégé Plugin, to identify hidden strict information. Performance of the algorithms is extensively analyzed exploiting both real world ontologies and artificial ones, demonstrating the feasibility of preferential reasoning under rational closure.

RAT-OWL exploits an alternative approach for computing rational closure, based on an encoding of the typicality operator into the standard DL, developed in [18] for $\mathcal{SHIQ}$. The rational closure construction requires a quadratic number of calls to an OWL reasoner (in the number of concepts, for the full modality, and in the number of typicality assertions in the KB, for the restricted modality). In future work we aim at extending our experimentation to further ontologies, such as those in [23], and at dealing with more expressive DLs. In this regard, we observe that establishing a correspondence between the rational closure construction and the minimal model semantics is still an open issue for expressive DLs including nominals and the universal role.

A further point to be considered is reasoning in stronger variants of the rational closure. It is well known that the rational closure does not allow independent handling of the inheritance of different defeasible properties of concepts. To overcome the limitation, several proposals have been developed. In [10] the lexicographic closure introduced by Lehmann [21] is extended to DLs, and in [19] a finer grained semantics, with several preference relations, is shown to correspond to a refinement of the rational closure in [17]. Moodley in [23] studies different kinds of closures and related algorithms, including algorithms for computing the lexicographic [10] and the relevant [7] closures, identifying the major bottlenecks for preferential reasoning in comparison with the rational closure.

In [2] a new non monotonic description logics $\mathcal{DL}^N$ has been proposed, which supports normality concepts and enjoys good computational properties. In particular, $\mathcal{DL}^N$ preserves the tractability of low complexity DLs, including $\mathcal{EL}^{\bot^{++}}$ and $DL\text{-}lite$ [5]. Inheritance of defeasible properties in $\mathcal{DL}^N$ is based on a notion of overriding, which builds over the rational closure in [9] to give preference to more specific defeasible inclusions with respect to less specific ones (and in this respect it has some similarity with the lexicographic closure). In future work we aim at exploring generalizations of the presented approach to deal with refinements of the rational closure, to get a stronger notion of entailment which overcomes the rational closure limitations.

Acknowledgements. We thank the anonymous referees for their helpful comments. The research has been supported by: the project INdAM-GNCS 2016 *"Ragionamento Defeasible nelle Logiche Descrittive"*; the project of Univ. Piemonte Orientale *"Defeasibility and interactions in ontological reasoning and description logics"*; and the project *"ExceptionOWL: Nonmonotonic Extensions of Description Logics and OWL for defeasible inheritance with exceptions"* Univ. Torino - Compagnia di San Paolo 2014.

References

1. Baader, F., Hollunder, B.: Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. Journal of Automated Reasoning (JAR) 15(1), 41–68 (1995)
2. Bonatti, P.A., Faella, M., Petrova, I., Sauro, L.: A new semantics for overriding in description logics. Artificial Intelligence 222, 1–48 (2015)
3. Bonatti, P.A., Faella, M., Sauro, L.: Defeasible inclusions in low-complexity dls. Journal of Artificial Intelligence Research (JAIR) 42, 719–764 (2011)
4. Bonatti, P.A., Lutz, C., Wolter, F.: The complexity of circumscription in dls. Journal of Artificial Intelligence Research (JAIR) 35, 717–773 (2009)
5. Bonatti, P.A., Petrova, I.M., Sauro, L.: Optimizing the computation of overriding. In: Arenas, M., Corcho, Ó., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P.T., Dumontier, M., Heflin, J., Thirunarayan, K., Staab, S. (eds.) The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9366, pp. 356–372. Springer (2015)
6. Bonatti, P.A., Sauro, L.: On the logical properties of the nonmonotonic description logic dl^{n}. Artif. Intell. 248, 85–111 (2017)
7. Casini, G., Meyer, T., Moodley, K., Nortje, R.: Relevant closure: A new form of defeasible reasoning for description logics. In: JELIA 2014. pp. 92–106. LNCS 8761, Springer (2014)
8. Casini, G., Meyer, T., Varzinczak, I.J., , Moodley, K.: Nonmonotonic Reasoning in Description Logics: Rational Closure for the ABox. In: Proc. DL 2013. pp. 600–615 (2013)
9. Casini, G., Straccia, U.: Rational Closure for Defeasible Description Logics. In: Janhunen, T., Niemelä, I. (eds.) Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA 2010). Lecture Notes in Artificial Intelligence, vol. 6341, pp. 77–90. Springer, Helsinki, Finland (September 2010)
10. Casini, G., Straccia, U.: Lexicographic closure for defeasible description logics. In: Proc. Australasian Ontology Workshop. pp. 28–39 (2012)
11. Casini, G., Straccia, U.: Defeasible Inheritance-Based Description Logics. Journal of Artificial Intelligence Research (JAIR) 48, 415–473 (2013)
12. Donini, F.M., Nardi, D., Rosati, R.: Description logics of minimal knowledge and negation as failure. ACM Transactions on Computational Logic (ToCL) 3(2), 177–225 (2002)
13. Donini, F.M., Massacci, F.: Exptime tableaux for $\mathcal{ALC}$. Artificial Intelligence 124(1), 87–138 (2000)
14. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: ALC+T: a preferential extension of description logics. Fundamenta Informaticae 96, 341–372 (2009)
15. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: A NonMonotonic Description Logic for Reasoning About Typicality. Artificial Intelligence 195, 165–202 (2013), `http://dx.doi.org/10.1016/j.artint.2012.10.004`

16. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Minimal model semantics and rational closure in description logics. In: DL 2013, 26th International Workshop on Description Logics. CEUR Workshop Proceedings, vol. 1014, pp. 168–180. CEUR-WS.org (2013)
17. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Semantic characterization of Rational Closure: from Propositional Logic to Description Logics. Artificial Intelligence 226, 1–33 (2015)
18. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Rational closure in $\mathcal{SHIQ}$. In: DL 2014, 27th International Workshop on Description Logics. CEUR Workshop Proceedings, vol. to appear. CEUR-WS.org (2014)
19. Gliozzi, V.: Reasoning about multiple aspects in rational closure for dls. In: Adorni, G., Cagnoni, S., Gori, M., Maratea, M. (eds.) AI*IA 2016: Advances in Artificial Intelligence - XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 - December 1, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10037, pp. 392–405. Springer (2016)
20. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence 44(1-2), 167–207 (1990)
21. Lehmann, D.J.: Another perspective on default reasoning. Ann. Math. Artif. Intell. 15(1), 61–82 (1995)
22. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence 55(1), 1–60 (1992)
23. Moodley, K.: Practical Reasoning for Defeasible Description Logics. PhD Thesis, University of Kwazulu-Natal (2016)
24. Moodley, K., Meyer, T., Sattler, U.: DIP: A defeasible-inference platform for OWL ontologies. In: 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014. CEUR Workshop Proceedings, vol. 1193, pp. 671–683. CEUR-WS.org (2014)
25. Straccia, U.: Default inheritance reasoning in hybrid kl-one-style logics. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI 1993). pp. 676–681. Morgan Kaufmann, Chambéry, France (August 1993)