An Efficient Reasoner for Description Logics of Typicality and Rational Closure*

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. In this work we present RAT-OWL, a Protégé 4.3 Plugin for reasoning about typicality in preferential Description Logics. RAT-OWL 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 $\mathbf{T}$: $\mathbf{T}(C) \sqsubseteq D$ represents that "typical Cs are also Ds". We show experimental results, indicating that the performances of RAT-OWL are promising.

1 Introduction

Nonmonotonic extensions of DLs have been investigated since the early 90s [4, 1, 3, 11, 10, 23, 2] in order to extend the standard languages to reason about prototypical properties and defeasible inheritance. A simple but powerful nonmonotonic extension of DLs is proposed in [12, 13, 15]: 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* [18] or *rational logic* [20]. We focus on the Description Logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ introduced in [15]. In this logic one can express defeasible inclusions such as "normally, athletes are not fat people": $\mathbf{T}(Athlete) \sqsubseteq \neg Fat$. As a difference with standard DLs, one can consistently express exceptions and reason about defeasible inheritance as well. 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: $\{ SumoWrestler \sqsubseteq Athlete$, $\mathbf{T}(Athlete) \sqsubseteq \neg Fat$, $\mathbf{T}(SumoWrestler) \sqsubseteq Fat \}$. In the Description Logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ standard models are extended by 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". The extension of $\mathbf{T}(C)$ is defined as the set of C elements that are minimal with respect to such preference relation. Intuitively, the semantics of the typicality operator $\mathbf{T}$ is strongly related to the one of rational logic $\mathbf{R}$ introduced by Kraus, Lehmann and Magidor [20]. This allows the typicality operator to inherit well-established properties of nonmonotonic reasoning: as an example, the property known as *specificity*, namely the choice of according preference to more specific information in case of conflicts

* This work has been partially supported by the projects "ExceptionOWL" (Universitá di Torino and Compagnia di San Paolo) and "Ragionamento Defeasible nelle Logiche Descrittive" (INdAM GNCS).

among inherited properties, results "built-in" in the approach. In the above example, if one knows that Paul is a typical sumo wrestler and, therefore, an athlete, then the logic allows to infer $\mathbf{T}(Fat)(paul)$, i.e. that Paul is fat, giving preference to the most specific information (sumo wrestler with respect to athlete).

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. As a consequence, unless a KB contains explicit assumptions about typicality of individuals, there is no way of inferring defeasible properties about them. Furthermore, the inclusion $\mathbf{T}(SumoWrestler \sqcap Blond) \sqsubseteq Fat$ 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 [15] the authors have introduced a nonmonotonic extension of the logic $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ based on a minimal model semantics, corresponding to a notion of *rational closure* as defined in [20] for propositional logic. Intuitively, the idea is to restrict our consideration to (canonical) models that maximize typical instances of a concept when consistent with the knowledge base. The resulting logic, called $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$, is based on a preference relation among $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ models and a notion of *minimal entailment* restricted to models that are minimal with respect to such preference relation. The rational closure construction proposed retains the same complexity of the underlying description logic: 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 paper we do not deal with the rational closure with respect to the ABox, developed in [15].

In this work we introduce RAT-OWL, a Protégé 4.3 Plugin for reasoning about typicality in the logic $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$. RAT-OWL relies on a polynomial encoding of $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ in standard $\mathcal{ALC}$ introduced in [14], based on the definition of the typicality operator $\mathbf{T}$ in terms of a Gödel-Löb modality $\Box$ as follows: $\mathbf{T}(C)$ is defined as $C \sqcap \Box\neg C$ where 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.

2 Description Logics of Typicality

In this section we recall the DLs of typicality, starting from 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 $\mathbf{T}$ [12]. 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 $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is 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: $\Delta^{\mathcal{I}}$ is the domain; $<$ is an irreflexive, transitive and modular (for all $x, y, z \in \Delta^{\mathcal{I}}$, if $x < y$ then either $x < z$ or $z < y$) relation over $\Delta^{\mathcal{I}}$; $.^{\mathcal{I}}$ is the extension function that maps each concept C to $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each role R to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For concepts of $\mathcal{ALC}$, C^{I} is defined in the usual way. For the $\mathbf{T}$ operator, we have $(\mathbf{T}(C))^{\mathcal{I}} = Min_{<}(C^{\mathcal{I}})$, where $Min_{<}(S) = \{u : u \in S$ and $\nexists z \in S$ s.t. $z < u\}$. Furthermore, $<$ satisfies the $Well-Foundedness\ Condition$, 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 $y < x$.*

We adopt usual definitions of satisfiability of inclusions and assertions in a model $\mathcal{M} \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} F$, satisfiability of a knowledge base $\mathcal{M} \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} K$, and of derivability of inclusion/assertion from K ($K \models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} F$).

Definition 3 (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 4 (Rank of a concept $k_{\mathcal{M}}(C_R)$). *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 prove that, for any $\mathcal{M}$, we have that $\mathcal{M}$ satisfies $\mathbf{T}(C) \sqsubseteq D$ if and only if $k_{\mathcal{M}}(C \sqcap D) < k_{\mathcal{M}}(C \sqcap \neg D)$.

In order to reason in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$, in [14] the authors provide the following polynomial encoding in standard $\mathcal{ALC}$ of KB[3]. The idea on which the encoding is based exploits the definition of the typicality operator $\mathbf{T}$ in terms of a Gödel-Löb modality $\Box$ as follows: $\mathbf{T}(C)$ is defined 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.

Let KB=($\mathcal{T}, \mathcal{A}$) be a knowledge base where $\mathcal{A}$ does not contain positive typicality assertions 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 $\mathcal{A}' = \emptyset$. 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 $\mathbf{T}(C) \sqsubseteq D \in \mathcal{T}$, we add to $\mathcal{T}'$ the inclusion $C \sqcap \Box_{\neg C} \sqsubseteq D$. In order to capture the properties of $\Box$ modality, a new role R is introduced to represent the relation $<$ in preferential models, and the following inclusions are introduced in $\mathcal{T}'$:

[3] The results provided in [14] are extended to the more expressive logic $\mathcal{SHIQ}$.

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

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 the inclusions, 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 C'_L and C'_R, assuming the size of C_L and C_R be polynomial in the size of KB.

Given the above encoding, in [14] it is shown that (we write KB $\models_{\mathcal{ALC}} F$ to mean that F holds in all $\mathcal{ALC}$ models of KB):

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

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 ExpTime-complete. ExpTime-hardness follows from the fact that $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ includes $\mathcal{ALC}$. In conclusion, the problem of deciding entailment in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ is ExpTime-complete.

Although the typicality operator $\mathbf{T}$ itself is nonmonotonic (i.e. $\mathbf{T}(C) \sqsubseteq D$ does not imply $\mathbf{T}(C \sqcap E) \sqsubseteq D$), 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 the same way, if we add to K the information that Jim is an athlete ($\mathit{Athlete}(jim)$), in $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ one cannot tentatively derive, in the absence of information to the contrary, that it is a typical athlete and therefore that he is not fat ($\mathbf{T}(\mathit{Athlete})(jim)$ and $\neg \mathit{Fat}(jim)$). In order to perform useful nonmonotonic inferences, in [15] the authors have strengthened the above semantics by restricting entailment to a class of minimal models. Intuitively, the idea is to restrict entailment to models that *minimize the untypical instances of a concept*. The resulting logic is called[4] $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$ and it corresponds to a notion of *rational closure* on top of $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$. Such a notion is a natural extension of the rational closure construction provided in [20] for the propositional logic.

Given a *query*, that is an inclusion of the form $C_L \sqsubseteq C_R$, we want to check whether it is entailed from a given knowledge base. First of all, we define notions of *exceptionality* of concepts and inclusions.

Definition 5 (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 6. *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 $C\}$.*

[4] We baptize the logic in this way here, for readability purposes.

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 7 (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.*

Consider the least $n \geq 0$ such that, for all $m > n, \mathcal{T}_m = \mathcal{T}_n$ or $\mathcal{T}_m = \emptyset$. Then 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 to define the rational closure of a knowledge base K with respect to the TBox.

Definition 8 (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 $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$ relies on minimal rational models that minimize the *rank of domain elements*. Informally, given two models of 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 KB, we say that $\mathcal{M}$ is a minimal model of KB with respect to $<$ if it is a model satisfying KB and there is no $\mathcal{M}'$ model satisfying KB such that $\mathcal{M}' < \mathcal{M}$. We further need to restrict our attention to *canonical models*. The intuition is that a canonical model contains all the individuals that enjoy properties that are consistent with the knowledge base. We consider all the sets of concepts $\{C_1, C_2, \ldots, C_n\} \subseteq \mathcal{S}$ that are *consistent* with KB, i.e., such that KB $\not\models_{\mathcal{ALC}^{\mathbf{R}}\mathbf{T}} C_1 \sqcap C_2 \sqcap \cdots \sqcap C_n \sqsubseteq \bot$, where $\mathcal{S}$ is the set of all the concepts (and subconcepts) occurring in KB together with their complements. Intuitively, a model $\mathcal{M}$ is a minimal canonical model of KB if it satisfies KB, it is minimal and it is canonical.

Query entailment in $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$ is then restricted to minimal canonical models: an inclusion $C_L \sqsubseteq C_R$ is entailed from K in $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$, written $K \models_{\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}} C_L \sqsubseteq C_R$, if $C_L \sqsubseteq C_R$ holds in all minimal canonical models of K with respect to $\mathcal{T}$.

In [15] a correspondence is shown between minimal model semantics and the construction of rational closure:

Theorem 1. *Let K=($\mathcal{T}, \mathcal{A}$) be a KB and $C_L \sqsubseteq C_R$ a query. We have that $C_L \sqsubseteq C_R \in \overline{\mathcal{T}}$ iff $C_L \sqsubseteq C_R$ holds in all minimal canonical models of K with respect to TBox.*

In [15] it is shown that the problem of deciding whether $\mathbf{T}(C) \sqsubseteq D \in \overline{\mathcal{T}}$ is in ExpTime, the same complexity upper bound of the underlying standard Description Logic $\mathcal{ALC}$.

3 Design of RAT-OWL

In this section we introduce RAT-OWL, which stands for **RAtional closure with Typicality in OWL**, and is intended to meet these needs. RAT-OWL[5] is a Protégé 4.3 Plugin

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

and it is written in Java and heavily uses OWL API 3.4 to manipulate OWL ontologies. It is based on the $\mathcal{ALC}^{\mathbf{R}}_{\text{RaCl}}\mathbf{T}$ logic, i.e. $\mathcal{ALC}^{\mathbf{R}}\mathbf{T}$ extended with rational closure of the TBox in order to perform nonmonotonic inferences. RAT-OWL makes use of the polynomial encoding into $\mathcal{ALC}$ described in the previous section. As an example, let the TBox contain: $1.\mathbf{T}(\mathit{Bird}) \sqsubseteq \mathit{Fly}, 2.\mathbf{T}(\mathit{Penguin}) \sqsubseteq \neg \mathit{Fly}, 3.\mathit{Penguin} \sqsubseteq \mathit{Bird}$. Its encoding[6] $\mathcal{T}'$ 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:

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 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. $\mathbf{T}(\mathit{Bird})$ **EquivalentTo** $(Bird$ **and** $Bird1)$
2. $NotBird1$ **EquivalentTo** $($**not** $(Bird1))$
3. $Bird1$ **SubClassOf** $(R$ **only** $($**not** $Bird$ **and** $Bird1))$
4. $NotBird1$ **SubClassOf** $(R$ **some** $(\mathbf{T}(\mathit{Bird})))$
5. $\mathbf{T}(\mathit{Bird})$ **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. Figure 1 illustrates the plugin interface. On the left-side of the window there is the hierarchy. On the right side there is the **Rational Closure Query View** where the user can write both

[6] In this implementation the auxiliary concept of a concept *C* is called *C*1 instead of $\Box\neg C$.

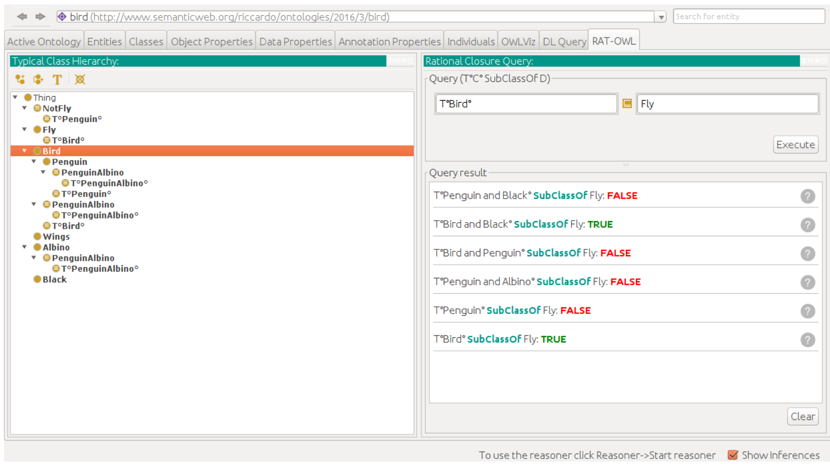

Fig. 1. RAT-OWL tab

classical queries such as $C \sqsubseteq D$ and typical 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 8, $\mathbf{T}(C) \sqsubseteq D$ 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.

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, namely the ontology that all levels have in common, then, for each level, only exceptional axioms $\mathcal{E}(E_i)$ are stored in *subsets* and not (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 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 typical query such as $\mathbf{T}(C) \sqsubseteq D$, as can be seen in Listing 1.2, if $rank(C)$ was already calculated in the rational closure *rankMap* it is not calculated again. This is the case, for instance, when $\mathbf{T}(C)$ is already contained in the KB. Furthermore, it is evident that $rank(C \sqcap \neg D)$ is at least $rank(C)$ so again we do not have to start necessarily from level 0.

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*, with *i* ∈ *[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 Performance 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 strong inclusions transformed into defeasible ones, in our case $C \sqsubseteq D$ is transformed into $\mathbf{T}(C) \sqsubseteq D$, 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 underlying 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, probably because each query is a subsumption $\mathbf{T}(C) \sqsubseteq D$ where $\mathbf{T}(C)$ does not belong to the ontology. Furthermore, for each parameter setting, rational closure was built up in two modalities: full and restricted. With the first one, for each concept C in the ontology, typical or not, its *rank* is computed, while with the second one only the *rank* of concepts C such that $\mathbf{T}(C)$ already exists in the ontology is computed. Figures 2 and 3 report the time needed with the full modality. We can observe

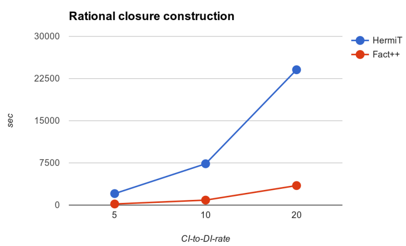

Fig. 2. HermiT and Fact++ rational closure construction time (15% DA-rate, full)

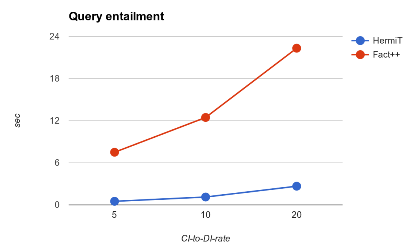

Fig. 3. HermiT and Fact++ query entailment time (15% DA-rate, full)

DA-rate	Rational closure	Query	DA-rate	Rational closure	Query
05%	1642.60	8.87	05%	445.87	18.89
10%	3134.58	12.73	10%	700.56	26.38
20%	3378.90	22.38	20%	905.48	39.57
25%	3112.81	22.89	25%	1046.19	51.09
30%	4213.91	23.63	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	CI-to-DI-rate	Rational closure	Query
05%	200.78	7.51	05%	37.81	14.95
10%	887.92	12.47	10%	191.00	26.25
15%	2520.91	17.42	15%	692.46	35.67
20%	3473.62	22.34	20%	2048.55	50.80
25%	8316.73	39.88	25%	6812.30	81.15

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

how with the restricted modality query entailment time increases while rational closure construction time decreases, in Table 1 by a factor of 2 and in Table 2 by a factor of 7.

Thus, for practical application of the rational closure of the TBox, the restricted modality may be preferred when the ontology is very complex.

Results in Table 2 show also how the increasing number of typicality axioms negatively affects the execution time as expected, as a matter of fact rational closure construction values are higher than the ones in Table 1. 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 good and overall lower than those in [5]. This witnesses that the performance of RAT-OWL is promising. It has to be noted, however, that the approach in [5] allows for a more sophisticated treatment of inheritance and overriding w.r.t. rational closure, which does not allow an independent treatment of different defeasible properties of a concept.

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 [20] 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 $\mathcal{ALC}$ [8, 10, 7, 15, 21]. 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 in [8, 7] is in the notion of exceptionality: our definition exploits preferential entailment, while [8, 7] directly use entailment in $\mathcal{ALC}$ over a materialization of the KB. In [22] a Defeasible-Inference Platform for OWL Ontologies has been proposed for the rational closure in [7], and in [21] a new algorithm for computing rational closure of TBox has been developed 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 analyzed on real-world ontologies in which defeasible inclusions have been injected, as well as on artificial ones, demonstrating the feasibility of preferential reasoning under rational closure. RAT-OWL exploits an alternative approach for computing the rational closure, based on an encoding of the typicality opertator in the standard DL, developed in [16] for $\mathcal{SHIQ}$. The rational closure construction requires a quadratic number of calls (in the number of typicality assertions in the KB) to an $\mathcal{ALC}$ reasoner. In future work we aim at extending our experimentation to further ontologies, such as those considered in [21], and to 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 to deal independently with the inheritance of different defeasible properties of concepts. To overcome the limitation, in [9] the lexicographic closure introduced by Lehmann [19] is extended to DLs, and in [17] a finer grained semantics, with several preference relations, is shown to correspond to a refinement of the rational closure in [15]. Moodley in [21] studies different kinds of closures and related algorithms, including algorithms for computing the lexicographic [9] and the relevant [6] 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 $\mathcal{EL}^{\perp ++}$ 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 [8] to give preference to more specific defeasible inclusions with respect to less specific ones. In future work we aim at exploring generalizations of the approach presented in this paper to deal with refinements of the rational closure which overcome the rational closure limitations.

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. 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)
7. 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)
8. 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)
9. Casini, G., Straccia, U.: Lexicographic closure for defeasible description logics. In: Proc. Australasian Ontology Workshop. pp. 28–39 (2012)
10. Casini, G., Straccia, U.: Defeasible Inheritance-Based Description Logics. Journal of Artificial Intelligence Research (JAIR) 48, 415–473 (2013)
11. 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)
12. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: ALC+T: a preferential extension of description logics. Fundamenta Informaticae 96, 341–372 (2009)
13. 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
14. 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)
15. 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)
16. 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)
17. 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)

18. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence 44(1-2), 167–207 (1990)
19. Lehmann, D.J.: Another perspective on default reasoning. Ann. Math. Artif. Intell. 15(1), 61–82 (1995)
20. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence 55(1), 1–60 (1992)
21. Moodley, K.: Practical Reasoning for Defeasible Description Logics. PhD Thesis, University of Kwazulu-Natal (2016)
22. 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)
23. 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)