DIP: A Defeasible-Inference Platform for OWL Ontologies

Thomas Meyer, Kody Moodley and Uli Sattler

Centre for Artificial Intelligence Research
CSIR Meraka Institute and UKZN, South Africa and
University of Manchester, United Kingdom
tmeyer@csir.co.za, {moodleyk,sattler}@cs.man.ac.uk

Abstract. The preferential approach to nonmonotonic reasoning was consolidated in depth by Krause, Lehmann and Magidor (KLM) for propositional logic in the early 90's. In recent years, there have been efforts to extend their framework to Description Logics (DLs) and a solid theoretical foundation has already been established towards this aim. Despite this foundation, and the fact that many of the desirable aspects of the approach generalise favourably to certain DLs, implementations thereof remain unpublished. We present a defeasible-reasoning system for OWL ontologies demonstrating that we need not devise new decision procedures for certain preferential DLs. Our reasoning procedures are composed purely of classical DL decision steps which allows us to immediately hinge upon existing OWL and DL systems for defeasible-reasoning tool support.

1 Introduction

The so-called *Preferential* or *KLM* approach [20] to nonmonotonic reasoning was introduced in the early 90's for propositional logic. In recent years, it has been shown that many of the desirable aspects of this approach can be generalised to certain fragments of first order logic such as the Description Logic (DL) $\mathcal{ALC}$ [14,6,13]. This preferential generalisation to $\mathcal{ALC}$ has some attractive attributes. Firstly, it was shown to facilitate an intuitive representation of defeasible statements (defaults) [13,6]. It also allows one to draw desirable defeasible conclusions [8, Section 3] which are as satisfactory as (if not superior to) the more well-known circumscriptive approaches [4,15]. But the most attractive properties, yet, of this logic are that it has a reasoning procedure which is composed purely of classical DL decision steps [8]; the worst case computational complexity stays the same as classical $\mathcal{ALC}$ ([14] and [9, Corollary 2]) and preliminary experiments show that the performance in practice is promising [8].

Despite these attributes, preferential approaches for defeasible reasoning in DLs have, largely, not been implemented. In fact, no implementation is published to our knowledge. This trend unfortunately carries over to other nonmonotonic approaches for DLs as well. Despite being the most published approach for nonmonotonic reasoning in DLs, circumscription does not have a well-known implementation. Hence, it is not surprising that tools for building and editing ontologies containing defeasible information are virtually absent in practice.

Preferential $\mathcal{ALC}$ has been shown to have decision procedures which reduce favourably to classical DL decision steps. This paper provides the good news that with surprisingly minor modifications, we are already able to hinge upon existing OWL and DL systems as tool support for defeasible reasoning and defeasible-ontology editing.

The system we are going to present, called Defeasible-Inference Platform (DIP), is based on a reasoning procedure for our preferential extension of $\mathcal{ALC}$. The theoretical foundation of this logic is well-founded and well documented [6,5] therefore we do not repeat much of the associated details thereof. Nevertheless, we will explain briefly the notion of defeasibility we subscribe to in this logic; the kinds of sentences we can express in this logic; and what knowledge bases (KBs) in this logic look like. We also briefly discuss the entailment question in such a logic and the unique fact that, in the preferential context, we have potentially *several* proposals for answering it. The aspects of the Protégé Ontology Editor and OWL standard which enable the simple integration of defeasible features are highlighted. Finally, we introduce our Defeasible-Inference Platform (DIP) as integrated into Protégé and conclude with a discussion about future work.

2 Preferential Reasoning

In classical DLs [1], the semantics is built upon first order interpretations. These interpretations vary on the elements which appear in the interpretation domain ($\Delta^{\mathcal{I}}$) and the manner in which we assign terms (defined by an interpretation function ($\cdot^{\mathcal{I}}$)) to these elements. In the preferential context, we introduce an additional component on which the interpretations can vary. This component represents the manner in which we order the elements of the domain, using a *partial* ordering ($\prec_{\mathcal{I}}$). Interpretations with this additional component are known as *preferential* interpretations. In order to be able to *rank* the elements of our domain, we need to specify that the partial order be *modular* [6, Definition 1]. This is so that we are able to assign suitable *ranks* to elements that are incomparable in the partial order. Hence, preferential interpretations whose orderings are modular are known as *ranked* interpretations. The ordering component of a ranked interpretation allows one to interpret so-called *defeasible* subsumption statements of the form $C \sqsubset\!\!\!\sim D$ (see Figure 1).

In contrast to standard DL subsumption ($C \sqsubseteq D$), which we read as "all C's are D's", the corresponding defeasible subsumption ($C \sqsubset\!\!\!\sim D$) is read as "the most *typical* C's are D's". It is the ordering on the elements in a ranked interpretation that allows us to identify or specify these typical elements under consideration. The semantic paradigm which this approach captures is very intuitive because it is one which we as humans often employ (albeit in an implicit way). Consider the following example:

Example 1. Suppose that Bob and John are mechanics. If we don't have any other information then as humans we may implicitly regard Bob and John as *typical*

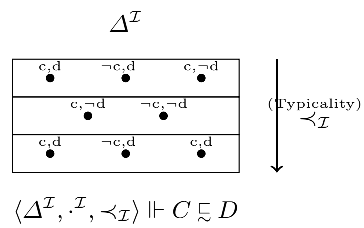


Fig. 1. Satisfaction of a defeasible subsumption by a ranked interpretation.

mechanics and assign to them properties that a typical mechanic may possess. For example we may conclude that Bob and John both work in a workshop. However, we may later discover that, while Bob works from a workshop, John is actually a *mobile* mechanic and only repairs machinery at the clients premises - which means he does not work from a workshop. One may say that Bob is more typical than John w.r.t. the property of possessing a workshop. Conversely, what this means is that John is more *exceptional* than Bob w.r.t. the same property. But what if we consider a different property of a typical mechanic? We may consider a typical mechanic to have one or more types of machinery that they *specialise* in. If we find that John indeed has a specialisation in motorboats but that Bob does not have a specialisation in any specific equipment types then we implicitly consider John to be more typical than Bob in this context. □

Example 1 demonstrates the need to consider *all* typicality orderings possible when constructing ranked interpretations of the knowledge we are reasoning about. We argue that in previous presentations of the preferential approach for DLs, there has not been enough clarity on how the approach deals with or combines *multiple* typicality orderings (as in Example 1). In Example 1 if we *only* have the constraint that typical mechanics work in a workshop then John has to be considered more exceptional than Bob in *any* ranked model thereof. Conversely, if we *only* have the constraint that typical mechanics have a specialisation then Bob is more exceptional than John. But what if we have to satisfy *both* constraints? Suppose our background knowledge is that typical mechanics work in workshops *and* that typical mechanics have at least one specialisation. Consider three of the ranked models of this knowledge in Figure 2. It is clear

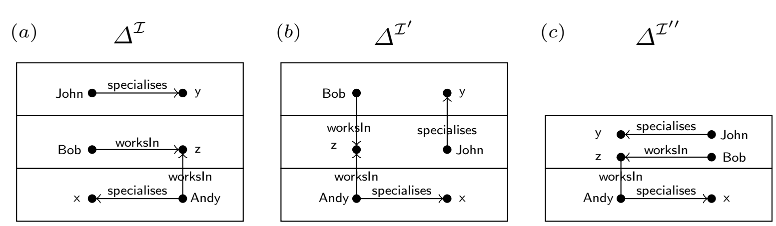


Fig. 2. Combining typicality orderings using ranked interpretations.

that if our background knowledge about mechanics is correct, then there must exist at least one typical mechanic out there who satisfies both our constraints. If there isn't then we obviously have to revise or retract our statements. Since Example 1 makes mention only of Bob and John, and both these individuals are missing one of the required properties, we have to conclude that there must be a third individual. We call him Andy and he is a very typical mechanic i.e. he possesses both required properties by working in a workshop and specialising in automobiles. Both Bob and John can then be seen as exceptional w.r.t. the prototypical mechanic Andy. But how do we decide who is more exceptional between Bob and John? The answer is that we don't have to because Andy satisfies our knowledge; Bob and John are exceptional to Andy so the exceptionality distinction between them does not matter ((a), (b) and (c) in Figure 2 are all valid models of our knowledge).

A strong advantage of preferential logics is the behaviour represented in Figures 1 and 2 where the ranked interpretations satisfy that the most typical C's (lowest in the ordering) are also D's, but still allows some C's that are not as typical (higher up in the ordering) to not be D's. This is the ability to gracefully cope with *exceptions* - which is something that standard DLs cannot. We find in many fields such as biology and medicine that it is very common to encounter information which holds in general but is fallible under exceptional circumstances. Given this setting, biologists and medical professionals still have to draw conclusions and make decisions based upon this information. Preferential DLs are developed for applications of this kind.

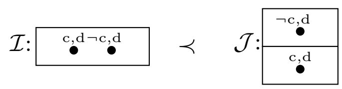
The state of the art within the framework of ranked interpretations is that we can reason with the terminological part of a *defeasible KB* [8] i.e. ABox approaches [7] are not mature. A defeasible KB is composed of a classical $\mathcal{ALC}$ TBox $\mathcal{T}$ and an $\mathcal{ALC}$ *DBox* $\mathcal{D}$ (set of defeasible inclusions of the form $C \sqsubsim D$).

Given a defeasible KB $\langle \mathcal{T}, \mathcal{D} \rangle$, the obvious first proposal for entailment of a defeasible inclusion $C \sqsubsim D$ would be to check in every ranked interpretation that satisfies every axiom in $\mathcal{T}$ and $\mathcal{D}$ and verify if $C \sqsubsim D$ is also satisfied there (a similar approach is used for entailment in standard DLs). However, it turns out that this proposal induces an entailment relation which is *monotonic* [5, Section 4] and defeats the purpose of our logic, which is supposed to enable the representation of potentially fallible statements that can be refuted upon the discovery of new information.

But, even though the proposal to consider *all* ranked models fails as mentioned above, it is still possible to narrow our view to a subset of these. The problem is that deciding which subset to focus on may be perceived as a subjective choice. Fortunately, in the context of propositional logic, KLM have argued extensively that it is not entirely subjective [20,18]. They delineated a series of logical properties that any nonmonotonic consequence relation should satisfy at bare minimum [20, Section 2.2]. They also pinpointed the smallest relation satisfying these properties coined the *Rational Closure* (RC) [20, Section 5].

A model-theoretic account of RC was also given by them which corresponds to considering the *minimal* ranked models [20, Section 5.7] as the base proposal for entailment. Minimal ranked models are defined by placing a partial ordering on the ranked models of the KB - this is in *addition* to the partial ordering on the elements of the domain (see Figure 3 for an example). The minimal ranked

$$\langle \mathcal{T}, \mathcal{D} \rangle = \langle \emptyset, \{C \sqsubset\!\!\!\sim D\} \rangle$$



$\mathcal{I}$ is a minimal ranked model for $\langle \mathcal{T}, \mathcal{D} \rangle$

Fig. 3. Ordering ranked models in pursuit of the minimal ones.

models in the partial order are those in which there is no element of the domain that can be moved to a more typical level in the strata (i.e. if it can be moved, then it is not possible without violating at least one axiom in the KB).

The logical properties that any nonmonotonic consequence relation should satisfy were shown to generalise well to the DL case ([5, Definition 4] and [6, Definition 2]). Several DL generalisations of RC have also been proposed [9,13,6,5]. Giordano et al. [13] gave the first generalisation of RC which corresponds in a natural way to the minimal ranked model semantics of KLM [20]. Our characterisation [5] was also shown to correspond to theirs.

The first attempt at a procedure for computing RC in the DL case was the effort of Casini and Straccia [9] for $\mathcal{ALC}$. This syntactic procedure was composed entirely of classical DL decision steps. A tableau calculus was presented for a preferential extension of $\mathcal{ALC}$ by Giordano et al. [14]. Notwithstanding, all existing procedures in the literature that are based on classical DL decision steps are variants of the syntactic procedure by Casini and Straccia [9].

The full technical details of our procedure including pseudocode has been presented [8]. We conclude our brief survey of preferential reasoning in DLs with an example illustrating the kinds of inferences we can draw with RC, the limitations of RC (the inferences that we would like to draw but cannot), and the additional inferences we can draw from recent extensions of RC such as the Lexicographic [19,10] and Relevant closures (submitted work).

Example 2. Consider the following defeasible KB:

$$\mathcal{T} = \left\{ \begin{array}{ll} 1.\ \mathsf{MRBCell} & \sqsubseteq \mathsf{ECell}, \\ 2.\ \mathsf{HRBCell} & \sqsubseteq \mathsf{MRBCell}, \\ 3.\ \mathsf{CamelRBCell} & \sqsubseteq \mathsf{MRBCell}, \\ 4.\ \exists hasShape.\mathsf{Circle} & \sqsubseteq \neg\exists hasShape.\mathsf{Oval} \end{array} \right\}$$

$$\mathcal{D} = \left\{ \begin{array}{ll} 1.\ \mathsf{ECell} & \sqsubset\!\!\!\sim\ \exists hasNucleus.\top, \\ 2.\ \mathsf{MRBCell} & \sqsubset\!\!\!\sim\ \neg\exists hasNucleus.\top, \\ 3.\ \mathsf{MRBCell} & \sqsubset\!\!\!\sim\ \exists hasShape.\mathsf{Circle}, \\ 4.\ \mathsf{HRBCell} \sqcap \exists hasCondition.\mathsf{EMH} & \sqsubset\!\!\!\sim\ \exists hasNucleus.\top, \\ 5.\ \mathsf{CamelRBCell} & \sqsubset\!\!\!\sim\ \exists hasShape.\mathsf{Oval} \end{array} \right\}$$

The defeasible KB $\langle \mathcal{T}, \mathcal{D} \rangle$ above, represents biological information describing that: eukaryotic cells (ECell) usually have a nucleus, mammalian red blood cells (MRBCell) are types of eukaryotic cells that usually don't possess a nucleus, human red blood cells (HRBCell) are also mammalian red blood cells but if they are affected by the *extramedullary hematopoiesis* [25] (EMH) medical condition then they usually contain a nucleus. In addition to the properties pertaining to nuclei, we also represent that mammalian red blood cells generally have a circular shape but the red blood cells of a camel (CamelRBCell), which are also mammalian, do not inherit this property (they are distinctly oval shaped) [21].

Using RC we are able to derive (retain) the intuitive inferences that: eukaryotic cells usually have a nucleus and even though mammalian red blood cells are considered eukaryotic, they are allowed to "break the tradition" and be devoid of a nucleus. In essence, mammalian red blood cells are recognised by RC as *exceptional* eukaryotic cells. RC also caters for *exceptions to exceptions* by noting that a human red blood cell that is infected with EMH is an exceptional mammalian red blood cell and is therefore allowed to possess a nucleus.

However, a limitation of RC is that it will not draw the reasonable inference that: human red blood cells (even if they are infected with EMH) should be circular in shape [19,10]. We can argue that this inference is reasonable to make because we know that mammalian red blood cells usually have a circular shape (Axiom 3 in $\mathcal{D}$), and that human red blood cells are mammalian (Axiom 2 in $\mathcal{T}$). The trouble is that RC sees human red blood cells with EMH as exceptional even though the reason for this has nothing to do with its shape (the reason is related to the property of possessing a nucleus). Together with the fact that RC does not permit inheritance of properties for exceptional elements, the desired inference is not allowed. In an analogous way, we cannot derive another desirable conclusion that a camel red blood cell should not possess a nucleus.

The Lexicographic and Relevant closures are *syntax-dependent* proposals that overcome the above limitations [19,10]. They do this by identifying the reasons for information to be considered exceptional in the KB (albeit in different ways). Relevant closure (submitted work) notably uses the notion of justifications [16,2] in this regard which further exploits the connection between nonmonotonic reasoning and belief revision [12]. In both these proposals, we are able to derive from Example 2 that human red blood cells infected with EMH are usually circular in shape and that camel red blood cells usually lack a nucleus. □

3 Protégé and OWL 2

In this section we briefly explain the relevant features of OWL 2 and Protégé that allow the expression of our notion of defeasible subsumption.

3.1 OWL

Since the advent of the Semantic Web vision [3], an important task towards achieving it has been to develop an appropriate language for constructing ontologies (the building blocks of the Semantic Web). The World Wide Web Consortium (W3C) formed the Web Ontology Working Group to develop such a suitable language. They came up with the Web Ontology Language (OWL) family of languages with DLs serving as their logical underpinning. OWL became a W3C recommendation in 2004 with its initial version dubbed OWL 1. The latest standard OWL 2 superseded OWL 1 as the W3C recommendation in 2009.

One of the new features in OWL 2 which makes it easy to express defeasible subsumption is the introduction of *OWL annotations*. These constructs allow one to attach meta-information to an entity of an OWL Ontology (be it a class, object property, individual name or axiom). One can therefore attach an annotation to a standard subsumption axiom in the ontology which allows a reasoning system to interpret this subsumption as a defeasible one. See Figure 4 for an OWL/XML rendering of the defeasible subsumption $\mathsf{MRBCell} \underset{\sim}{\sqsubset} \mathsf{ECell}$. OWL/XML is one of the various syntaxes that OWL ontologies can be serialised in (some notable alternatives are RDF/XML - `www.w3.org/TR/REC-rdf-syntax`, Manchester OWL Syntax - `www.w3.org/TR/owl2-manchester-syntax` and Turtle - `www.w3.org/TeamSubmission/turtle`).

```
<SubClassOf>
    <Annotation>
      <AnnotationProperty IRI="http://www.cair.za.net/defeasible"/>
          <Literal datatypeIRI="&xsd;boolean">true</Literal>
    </Annotation>
    <Class IRI="#ECell"/>
    <Class IRI="#MRBCell"/>
</SubClassOf>
```

Fig. 4. Defeasible subsumption in OWL syntax

3.2 Protégé

Protégé (`protege.stanford.edu`) is a software ontology editor capable of handling ontologies of various formats but predominantly tailored for OWL 2 ontologies. The latest version of the editor (Protégé 4.3) is capable of editing OWL 2 ontologies thanks to its use of the underlying Java-based API - the OWLAPI [11,17] - which is currently aligned with the OWL 2 standard of languages. Protégé makes it easy to load, create and manipulate OWL 2 ontologies and it has a plug-in friendly infrastructure which makes it ideal for extensibility. In fact, Protégé is itself highly modular and can be viewed as a set of interacting plugins. We are able to exploit its *rendering* components for defeasible-ontology editing support. For example, we are able to render defeasible subsumptions just as intuitively as standard subsumptions in the Protégé user interface (UI) (see Figure 5). There are two ways to add a defeasible subsumption to an ontology

MRBCell **UsuallySubClassOf** hasShape **some** Circle

Fig. 5. Defeasible subsumption rendering in Protégé

using DIP for Protégé: the first way is applicable when the user wishes to create a defeasible subsumption whose left-hand-side (LHS) is a concept name. This is done by selecting the LHS concept name in the class hierarchy and, in the corresponding (Class) Description window, adding a defeasible superclass for this concept name (see Figure 6). The second way is to manually type out the subsumption using the "UsuallySubClassOf" keyword (see Figure 5) in the General Class Axioms (GCI) window in Protégé. The GCI window can be accessed by going to Window→Views→Class Views in Protégé. The defeasible superclasses

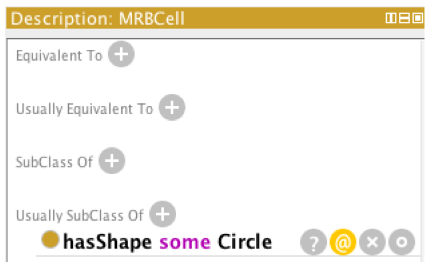


Fig. 6. Adding defeasible superclasses in Protégé

are interpreted as such through the use of OWL annotations as mentioned in Section 3.1. See Figure 7 for an example of an automatically generated defeasible annotation when the user creates a defeasible superclass in DIP for Protégé.

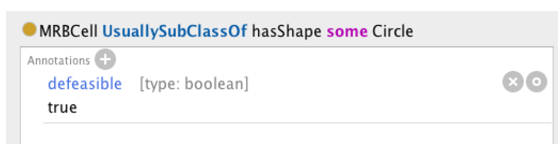


Fig. 7. Defeasible annotations in Protégé

4 DIP: Defeasible-Inference Platform

In addition to the revisions we have made to enable Protégé to represent defeasible information, we have developed an accompanying defeasible reasoner - a defeasible-inference platform (DIP)[1] - to be used in conjunction with this version of Protégé. See Figure 8 for a screenshot of the main interface.

We have implemented the Preferential, Rational and Lexicographic closures to date. Recall that Preferential closure corresponds to the entailment relation which considers *all* ranked models (described in Section 2). The basic architecture of the system is shown in Figure 9.

The basic workflow of DIP is as follows: the user supplies a *query* through the UI. The query is composed of a defeasible (or standard) subsumption axiom

[1] tinyurl.com/defeasible-inference-platform

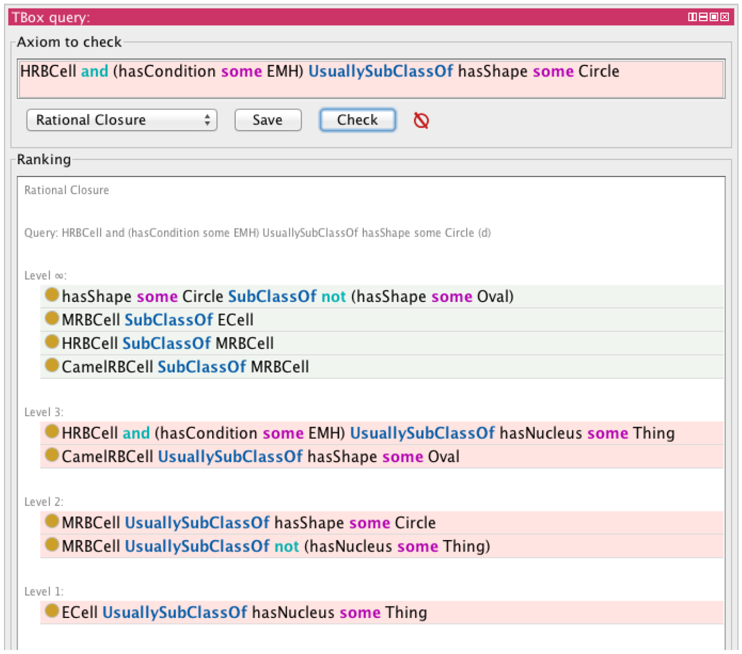


Fig. 8. DIP main interface in Protégé.

(entered via a text box) and a selected algorithm (i.e. Preferential, Rational or Lexicographic selected from a drop-down menu). The UI of our tool is modular and composed of a set of *view windows* and the *main toolbox*. The main toolbox is responsible for gathering the query as described above. Our query is called as such because we would like to answer "yes" or "no" depending on whether the entered axiom is contained in the selected closure of the loaded ontology. The "check" button is responsible for executing the query and a "save" button is provided for storing the query in the ontology file (again with the use of annotations) so that we can retrieve it even after the file is closed and reopened.

If it is the first time that we are initiating a query on the particular ontology version, then our defeasible reasoner has to combine the typicality orderings in the ranked models of the ontology to induce a *single* ordering on its axioms which we call the *ranking* [8]. The ranking is only computed once for each version of an ontology (see Figure 8 for an example). The ranking computation involves a number of classical DL entailment checks (in the worst case the number is factorial w.r.t. the number of axioms in the ontology) and these are carried out by the selected standard OWL reasoner (e.g. Pellet [23], HermiT [22] or FaCT++ [24]).

Once we obtain a ranking, the reasoner executes another series of classical entailment checks (in the worst case it is linear w.r.t. the number of axioms in the ranking) to determine if the query is contained in the selected closure. While the ontology is not changed, it is only these last checks that have to be performed

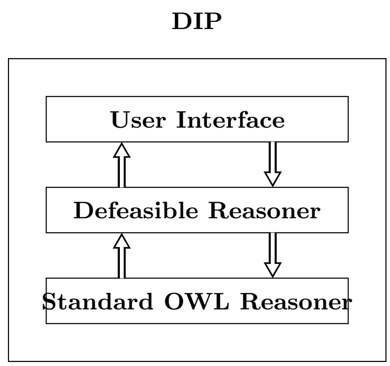


Fig. 9. Basic architecture of DIP.

for all subsequent queries we wish to answer. From a UI perspective, when the result of a query is negative (the specified axiom is *not* in the selected closure of the ontology) then we indicate this by a red symbol in the main toolbox (see Figure 8) and conversely, when the result is positive (the specified axiom is in the selected closure of the ontology) we indicate this with a green symbol (see Figure 10). DIP has included various view windows for convenience. There is

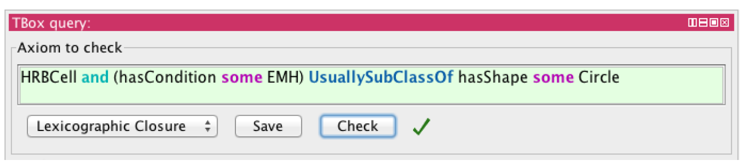


Fig. 10. Positive result in DIP for Protégé.

a window for displaying the defeasible subsumptions (and a separate one for displaying the standard subsumptions). There is also a display for the set of queries that the user has stored for future use. We allow the user to conveniently convert any subsumption axiom between defeasible and standard by means of a circular grey toggle button - labelled with the letter "d" (for defeasibility). See Figure 11 for a screenshot of these views.

DIP does not yet make use of the Protégé OWL reasoner interface like traditional OWL reasoners such as Pellet, HermiT and FaCT++. This is because we currently do not have highly optimised procedures for performing non-standard reasoning tasks such as *classification*. Classification is the task of computing all subset/superset relationships between every pair of class names in the ontology. In the preferential context, this task is not straightforward for various reasons. One reason is that properties which we take for granted in the monotonic case, such as transitivity, do not hold in general. Transitivity is broken by noting that: just because typical mechanics are male and typical males are tall does not necessarily mean that typical mechanics are tall. I.e. we can express that a

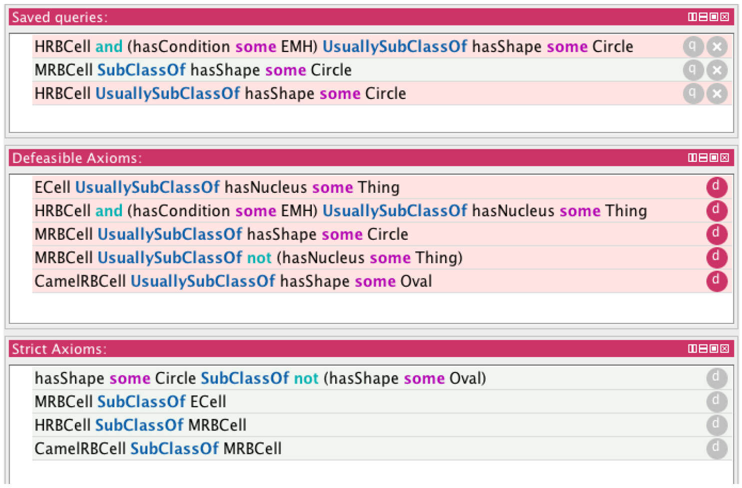


Fig. 11. Views for defeasible, strict and query axioms in DIP for Protégé.

typical mechanic should be male but we cannot constrain typical mechanics to necessarily be *typical* males in our language of defeasible subsumptions.

5 Conclusions and Future Work

We have given a brief survey of the intuitions behind the preferential approach to defeasible reasoning in DLs. We have pointed out that it gives intuitive defeasible conclusions back; in the case of preferential $\mathcal{ALC}$, it does so with procedures that reduce to classical DL decision steps and without increasing the computational complexity of classical $\mathcal{ALC}$. Our procedures also have performance results which are promising [8]. Further positive news is that existing OWL and DL tools for ontology editing can be used to represent our notion of defeasible subsumption in OWL 2 ontologies. We have also presented a preliminary defeasible-reasoning system (DIP) that can be used with such ontologies. The reasoner implements two initial proposals for nonmonotonic entailment: the Rational and Lexicographic closures. On the theoretical front, there are still various avenues to investigate: ABox reasoning in the preferential framework for $\mathcal{ALC}$, adapting the framework to more expressive logics, incorporating defeasible notions in other DL constructs and other extensions of Rational and Lexicographic closure. On the practical front we plan to develop optimisations for our procedures and investigate their practical performance on non-synthetic ontologies.

Acknowledgements. This work is based upon research supported by the National Research Foundation (NRF). Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF do not accept any liability in regard thereto. Kody Moodley is a Commonwealth Scholar, funded by the UK government.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. F. Baader and R. Peñaloza. Axiom pinpointing in general tableaux. *Journal of Logic and Computation*, 20(1):5–34, 2010.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. In *Scientific American*. Scientific American, May, 2001.
4. P. A. Bonatti, C. Lutz, and F. Wolter. The complexity of circumscription in DLs. *Journal of Artificial Intelligence Research*, 35:717–773, 2009.
5. K. Britz, G. Casini, T. Meyer, K. Moodley, and I. J. Varzinczak. Ordered Interpretations and Entailment for Defeasible Description Logics. Technical report, CAIR, CSIR Meraka and UKZN, South Africa, 2013.
6. K. Britz, T. Meyer, and I. Varzinczak. Semantic foundation for preferential description logics. In *Proc. of the Australasian Joint Conference on Artificial Intelligence*, pages 491–500. Springer, 2011.
7. G. Casini, T. Meyer, K. Moodley, and I. Varzinczak. Nonmonotonic reasoning in Description Logics. rational closure for the ABox. In *Proc. of DL*, 2013.
8. G. Casini, T. Meyer, K. Moodley, and I. J. Varzinczak. Towards practical defeasible reasoning for description logics. In *Proc. of DL*, 2013.
9. G. Casini and U. Straccia. Rational closure for defeasible description logics. In *Proc. of JELIA*, pages 77–90, 2010.
10. G. Casini and U. Straccia. Lexicographic closure for defeasible description logics. In *Proc. of Australasian Ontology Workshop*, volume 969, 2012.
11. B. Cuenca-Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler. Cooking the semantic web with the OWL API. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.
12. P. Gärdenfors. *Belief revision*, volume 29. Cambridge University Press, 2003.
13. L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Minimal model semantics and rational closure in description logics. In *Proc. of DL*, 2013.
14. L. Giordano, N. Olivetti, V. Gliozzi, and G. L. Pozzato. ALC + T: a preferential extension of description logics. *Fundamenta Informaticae*, 96(3):341–372, 2009.
15. S. Grimm and P. Hitzler. A preferential tableaux calculus for circumscriptive ALCO. In *Proc. of RR*, pages 40–54, 2009.
16. M. Horridge. *Justification based explanation in ontologies*. PhD thesis, the University of Manchester, 2011.
17. M. Horridge and S. Bechhofer. The OWL API: A Java API for OWL ontologies. *Semantic Web Journal*, pages 1–11, 2010.
18. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.
19. D. Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15:61–82, 1995.
20. D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55(1):1–60, 1992.
21. R. A. McPherson, W. H. Sawyer, and L. Tilley. Band 3 mobility in camelid elliptocytes: implications for erythrocyte shape. *Biochemistry*, 32(26):6696–6702, 1993.
22. R. Shearer, B. Motik, and I. Horrocks. HermiT: a highly efficient OWL reasoner. In *Proceedings of the Fifth International Workshop on OWL: Experiences and Directions (OWLED)*, 2008.

23. E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51 – 53, 2007.
24. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 292–297. Springer Berlin / Heidelberg, 2006.
25. R Verani, J Olson, and J. L. Moake. Intrathoracic extramedullary hematopoiesis: report of a case in a patient with sickle-cell disease-beta-thalassemia. *American journal of clinical pathology*, 73(1):133–137, 1980.