Defeasible reasoning in RDFS

(Extended Abstract)

Giovanni Casini[1,2], Umberto Straccia[1]

[1] ISTI - CNR, Pisa, Italy

[2] CAIR, University of Cape Town, South Africa

Abstract

For non-monotonic logics, the notion of Rational Closure (RC) is acknowledged as one of the main approaches. In this work we present an integration of RC within the triple language RDFS (Resource Description Framework Schema), which together with OWL 2 is a major standard semantic web ontology language. To do so, we start from ρdf, an RDFS fragment that covers the essential features of RDFS, and extend it to $\rho df_\perp$, allowing to state that two entities are incompatible/disjoint with each other. Eventually, we propose defeasible $\rho df_\perp$ via a typical RC construction allowing to state default class/property inclusions.

Keywords

RDFS, non-monotonic reasoning, defeasible reasoning, rational closure

1. Introduction

RDFS (*Resource Description Framework Schema*)[1] is a main standard semantic web ontology language that consists of triples (s, p, o) (denoting s is related via p with o). The introduction of non-monotonic formalisms in reasoning with ontologies is useful in particular to deal with situations in which some classes are exceptional and do not satisfy some typical properties of their super classes, as illustrated with the following example.

Example 1.1 (Running example)**.** *Consider the following facts (and an intuitive translation into RDFS, where* sc *is read as* "is a subclass of"*).*

- *Young people are usually happy;* (yP, sc, hP)
- *Drug users are usually unhappy;* (dU, sc, uhP)
- *Drug users are usually young;* (dU, sc, yP)
- *Controlled drug users are usually happy;* (cDU, sc, hP)
- *Controlled drug users are drug users;* (cDU, sc, dU)

We may consider then reasonable to conclude, for example, that controlled young drug users are usually happy.

Description Logics provide the logical foundation of formal ontologies of the semantic *Web Ontology Language* (OWL) family[2] and endowing them with non-monotonic features has been a main issue in the past 20 years [1, 2, 3, 4, 5, 6]. On the other hand, addressing non-monotonicity in the context of RDFS, has attracted in comparison little attention so far, and almost all approaches we are aware of implement non-monotonicity by adding a so-called rule-layer on top of RDFS; see *e.g.*, [7, 8, 9, 2, 10].

In the following, our aim is to show how to integrate Rational Closure (RC), one of the main constructions in non-monotonic reasoning [11], directly within the triple language RDFS. To to do so, we start from ρdf [12, 13], a minimal, but significant RDFS fragment that covers the essential features of RDFS, and then extend it to $\rho df_\perp$, allowing to state that two entities are incompatible/disjoint with each other. The results in this paper are presented more in detail in a technical report [14].

2. $\rho df_\perp$ Graphs

We rely on a fragment of RDFS, called *minimal* ρdf [12, Def. 15], that covers all main features of RDFS, and it is essentially the formal logic behind RDFS. The vocabulary is composed by two pairwise disjoint alphabets $\mathbf{U}$ and $\mathbf{L}$ denoting, respectively, *URI references* and *literals*, where a literal may be a *plain literal* (*e.g.*, a string) or a *typed literal* (*e.g.*, a boolean value) [15]. With $\mathbf{UL}$, the set of *terms*, we will denote the union of these sets. A ρdf-*triple* is of the form $\tau = (s, p, o) \in \mathbf{UL} \times \mathbf{U} \times \mathbf{UL}$.[3] We call s the *subject*, p the *predicate*, and o the *object*. A *graph* G is a set of triples. ρdf is characterised by the set of predicates $\{\mathsf{sp}, \mathsf{sc}, \mathsf{type}, \mathsf{dom}, \mathsf{range}\} \subseteq \mathbf{U}$, that can appear only as second elements in the triples. Informally, (i) (p, sp, q) means that property p is a *subproperty* of property q; (ii) (c, sc, d) means that class c is a *subclass* of class d; (iii) (a, type, b) means that a is of *type* b;

NMR 2022: 20th International Workshop on Non-Monotonic Reasoning, August 07–09, 2022, Haifa, Israel

giovanni.casini@isti.cnr.it (G. Casini); umberto.straccia@isti.cnr.it (U. Straccia)

0000-0002-4267-4447 (G. Casini); 0000-0001-5998-6757 (U. Straccia)

[1] http://www.w3.org/TR/rdf-schema/

[2] http://www.w3.org/TR/2009/REC-owl2-profiles-20091027

[3] As in [12], we allow literals for s.

(iv) (p, dom, c) means that the *domain* of property p is c; and (v) (p, range, c) means that the *range* of property p is c. We also recall that minimal ρdf does not consider so-called *blank* nodes [16, 12].

Concerning the semantics of ρdf [12], an *interpretation* is a tuple $\mathcal{I} = \langle \Delta_\mathsf{R}, \Delta_\mathsf{P}, \Delta_\mathsf{C}, \Delta_\mathsf{L}, \mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^\mathcal{I} \rangle$, where $\Delta_\mathsf{R}, \Delta_\mathsf{P}, \Delta_\mathsf{C}, \Delta_\mathsf{L}$ are the interpretation domains of $\mathcal{I}$, which are finite non-empty sets, and $\mathfrak{P}[\![\cdot]\!], \mathfrak{C}[\![\cdot]\!], \cdot^\mathcal{I}$ are the interpretation functions of $\mathcal{I}$. In particular: (i) Δ_R are the resources (the domain or universe of $\mathcal{I}$); (ii) Δ_P are property names (not necessarily disjoint from Δ_R); (iii) $\Delta_\mathsf{C} \subseteq \Delta_\mathsf{R}$ are the classes; (iv) $\Delta_\mathsf{L} \subseteq \Delta_\mathsf{R}$ are the literal values and contains $\mathbf{L} \cap V$; (v) $\mathfrak{P}[\![\cdot]\!]$ is a function $\mathfrak{P}[\![\cdot]\!] \colon \Delta_\mathsf{P} \to 2^{\Delta_\mathsf{R} \times \Delta_\mathsf{R}}$; (vi) $\mathfrak{C}[\![\cdot]\!]$ is a function $\mathfrak{C}[\![\cdot]\!] \colon \Delta_\mathsf{C} \to 2^{\Delta_\mathsf{R}}$; (vii) $\cdot^\mathcal{I}$ maps each $t \in \mathbf{UL} \cap V$ into a value $t^\mathcal{I} \in \Delta_\mathsf{R} \cup \Delta_\mathsf{P}$, and such that $\cdot^\mathcal{I}$ is the identity for plain literals and assigns an element in Δ_R to each element in $\mathbf{L}$.

An interpretation $\mathcal{I}$ satisfies a graph G if for each $(s, p, o) \in G$, $p^\mathcal{I} \in \Delta_\mathsf{P}$ and $(s^\mathcal{I}, o^\mathcal{I}) \in \mathfrak{P}[\![p^\mathcal{I}]\!]$, and moreover $\mathcal{I}$ satisfies a series of constraints related to the ρdf-predicates. For example, a constraint imposing that $\mathfrak{P}[\![\mathsf{sc}^\mathcal{I}]\!]$ is transitive over Δ_P indicates that the subclass relation sc must be transitive. We refer to [12, Def. 15] for the full definition of the satisfaction relation, and of the correspondent entailment relation.

Definition 2.1 (Entailment $\models_{\rho df_\perp}$)**.** *Given two graphs G and H, we say that G* entails *H, denoted $G \models_{\rho df} H$, if and only if every model of G is also a model of H.*

In [12] the reader can find also a deduction system, consistent and complete w.r.t. the ρdf entailment relation, that is based on rules, such as

$$\frac{(A, \mathsf{sc}, B), (B, \mathsf{sc}, C)}{(A, \mathsf{sc}, C)}$$

encoding the transitivity of sc.

Defeasible reasoning can be built only when faced with a conflict between the properties of a class and of a subclass. *e.g.*, in Example 1.1,"Drug users are usually unhappy" appears in conflict with "Controlled drug users are usually happy". ρdf is not expressive enough to model such conflicts. So, we need to introduce at least a notion of incompatibility, of *disjunctiveness* [17]. Hence we enrich the ρdf vocabulary with two new predicates, $\perp_\mathsf{c}$ and $\perp_\mathsf{p}$, representing incompatible information: $(c, \perp_\mathsf{c}, d)$ (resp., $(p, \perp_\mathsf{p}, q)$) indicates that the classes c and d (resp., the properties p and q) are disjoint. Of course we can further enrich the language allowing for logically stronger notions such as negation [18], but it is not necessary for the purpose of the present paper.

We call the new formalism, obtained by adding $\perp_\mathsf{c}$ and $\perp_\mathsf{p}$ to ρdf, $\rho df_\perp$. Some new constraints are added to the semantics of ρdf [14, Sect. 2.2]. Here are a few examples:

- if $(c, d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ then $c, d \in \Delta_\mathsf{C}$;
- If $(c, d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$, then $(d, c) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ (*sc-Symmetry*);
- If $(c, d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ and $(e, c) \in \mathfrak{P}[\![\mathsf{sc}^\mathcal{I}]\!]$, then $(e, d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ (*sc-Transitivity*);
- If $(c, c) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ and $d \in \Delta_\mathsf{C}$ then $(c, d) \in \mathfrak{P}[\![\perp_\mathsf{c}{}^\mathcal{I}]\!]$ (*c-Exhaustive*).

These new constraints are such to model relevant properties of disjointedness, and allow the definition of an entailment relation $\models_{\rho df_\perp}$. An important feature of $\rho df_\perp$ is also that it preserves the ρdf property that a graph is always satisfiable, avoiding the possibility of unsatisfiability and the *ex falso quodlibet* principle. This is in line with the ρdf semantics [12, 19]. From an inference system point of view, new derivation rules are added to the ρdf derivation system [14, Sect. 2.3]. The following are just a few examples:

$$\frac{(A, \perp_\mathsf{c}, B)}{(B, \perp_\mathsf{c}, A)} \;; \quad \frac{(A, \perp_\mathsf{c}, B), (C, \mathsf{sc}, A)}{(C, \perp_\mathsf{c}, B)} \;; \quad \frac{(A, \perp_\mathsf{c}, A)}{(A, \perp_\mathsf{c}, B)} \,.$$

The new derivation relation $\vdash_{\rho df_\perp}$ that we have defined is correct and complete w.r.t. the entailment relation $\models_{\rho df_\perp}$ [14, Th. 2.1]. Eventually, we say that a graph G has a *conflict* if, for some term t, either $G^s \vdash_{\rho df_\perp} (t, \perp_\mathsf{c}, t)$ or $G^s \vdash_{\rho df_\perp} (t, \perp_\mathsf{p}, t)$ holds. The intuitive meaning is that G has a conflict if we can derive for some term t that it is either an empty class, $(t, \perp_\mathsf{c}, t)$, or an empty predicate, $(t, \perp_\mathsf{p}, t)$.

Example 2.1 (Running example cont.)**.** *In Example 1.1 we could add the triple $(uhP, \perp_\mathsf{c}, hP)$ to indicate that 'being happy' and 'being unhappy' are incompatible. Notice that from $(uhP, \perp_\mathsf{c}, hP)$, (cDU, sc, hP), (cDU, sc, dU) and (dU, sc, uhP) we conclude $(cDU, \perp_\mathsf{c}, cDU)$, that is, that being a controlled drug user is incompatible with being a controlled drug user (that is, cDU should be an empty class). Analogously, from $(uhP, \perp_\mathsf{c}, hP)$, (dU, sc, yP), (yP, sc, hP) and (dU, sc, uhP) we conclude $(dU, \perp_\mathsf{c}, dU)$.*

3. Defeasible $\rho df_\perp$

Next we show how to model defeasible information. Here we consider defeasibility w.r.t. the predicates sc and sp only, and introduce the notion of *defeasible triple*:

$$\delta = \langle s, p, o \rangle \in \mathbf{UL} \times \{\mathsf{sc}, \mathsf{sp}\} \times \mathbf{UL} \,,$$

where $s, o \notin \rho df_\perp$. The intended meaning of *e.g.*, $\langle c, \mathsf{sc}, d \rangle$ is "Typically, an instance of c is also an instance of b". Analogously, $\langle p, \mathsf{sp}, q \rangle$ is read as "Typically, a pair related by p is also related by q".

Example 3.1 (Running example cont.). *In Example 1.1 the statements containing* 'usually' *can more correctly be modelled using defeasible triples, that is,* $\langle yP, \mathsf{sc}, hP\rangle, \langle dU, \mathsf{sc}, uhP\rangle, \langle dU, \mathsf{sc}, yP\rangle$ *and* $\langle cDU, \mathsf{sc}, hP\rangle$.

There are various ways of reasoning in a defeasible framework. Here we take under consideration RC [11], since, despite having some limits from the inferential point of view [20], it is a main inference relation in conditional reasoning on top of which we can define other interesting forms of entailment [20, 21, 22].

We give here only a short overview of the reasoning procedure, inviting the reader to check [14] for a comprehensive presentation. Given a defeasible graph G and a query $\langle s, p, o\rangle$, we decide whether $\langle s, p, o\rangle$ is in the RC of G through a two-step procedure:

1. We rank all the defeasible triples in G, considering the potential conflicts and the relative logical specificity of the first elements of the triples. We give priority (that is, a higher rank) to more specific triples. To check the presence of potential conflicts in a graph, we translate all the defeasible triples into the correspondent $\rho df_\bot$ triples, that is, we create a new $\rho df_\bot$ graph in which every defeasible $\langle s, p, o\rangle$ is substituted by (s, p, o).

Example 3.2 (Running example cont.). *In Example 2.1 we have seen that from the* $\rho df_\bot$ *version of our graph we obtain* $(cDU, \bot_{\mathsf{c}}, cDU)$ *and* $(dU, \bot_{\mathsf{c}}, dU)$. *From this we conclude that all the defeasible triples with* cDU *or* dU *as first element (e.g.,* $\langle cDU, \mathsf{sc}, hP\rangle$ *and* $\langle dU, \mathsf{sc}, uhP\rangle$*) have priority (a higher rank) w.r.t. the other defeasible triples. That is,* $\langle yP, \mathsf{sc}, hP\rangle$ *has rank 0, while the other defeasible triples are exceptional. We then reiterate the procedure considering only the exceptional triples and the* $\rho df_\bot$*-triples, that is,* $\{\langle dU, \mathsf{sc}, uhP\rangle, \langle dU, \mathsf{sc}, yP\rangle, \langle cDU, \mathsf{sc}, hP\rangle\} \cup \{(cDU, \mathsf{sc}, dU), (hP, \bot_{\mathsf{c}}, uhP)\}$. *Translating the defeasible triples into* $\rho df_\bot$*-triples, the only conflict we can still derive is* $(cDU, \bot_{\mathsf{c}}, cDU)$*, hence we have that* $\langle dU, \mathsf{sc}, uhP\rangle, \langle dU, \mathsf{sc}, yP\rangle$ *have rank 1, while* $\langle cDU, \mathsf{sc}, hP\rangle$ *is exceptional. From* $\{\langle cDU, \mathsf{sc}, hP\rangle\} \cup \{(cDU, \mathsf{sc}, dU), (hP, \bot_{\mathsf{c}}, uhP)\}$ *we cannot derive anymore* $(cDU, \bot_{\mathsf{c}}, cDU)$*, hence* $\langle cDU, \mathsf{sc}, hP\rangle$ *has rank 2 and we have finished the ranking of the graph.*

Note that, given a graph G, the ranking procedure needs to be done once and for all.

2. Given a query $\langle s, \mathsf{sc}, o\rangle$ (resp., $\langle s, \mathsf{sp}, o\rangle$), we check the rank of s, *i.e.*, we check which is the lowest rank in which we do not derive $(s, \bot_{\mathsf{c}}, s)$ (resp., $(s, \bot_{\mathsf{p}}, s)$), and then we check whether we can derive (s, sc, o) (resp., (s, sp, o)) considering only the defeasible triples with at least such a rank.

Example 3.3 (Running example cont.). *We wonder whether* $\langle cDU, \mathsf{sc}, uhP\rangle$ *is in the RC of our graph. This triple is interesting because it would be derivable in the monotonic* $\rho df_\bot$*-graph we have considered up to Exmple 2.1, but it is undesirable since we are aware that* $\langle cDU, \mathsf{sc}, hP\rangle$ *and that 'Drug users are* usually *happy', that is a defeasible statement. If we consider our entire graph, we already know (Example 3.2) that* cDU *is exceptional, that is, substituting the defeasible triples with their* $\rho df_\bot$ *counterparts, we obtain* $(cDU, \bot_{\mathsf{c}}, cDU)$. *The same if we consider the graph obtained eliminating all the defeasible triples of rank 0. Only once we eliminate also the triples of rank 1, and we consider only the graph* $\{\langle cDU, \mathsf{sc}, hP\rangle\} \cup \{(cDU, \mathsf{sc}, dU), (hP, \bot_{\mathsf{c}}, uhP)\}$*, we are not able to derive* $(cDU, \bot_{\mathsf{c}}, cDU)$ *anymore. That is, we do not have a conflict anymore on* cDU. *Our query* $\langle cDU, \mathsf{sc}, uhP\rangle$ *will be decided considering only this portion of the original graph:* $\{\langle cDU, \mathsf{sc}, hP\rangle\} \cup \{(cDU, \mathsf{sc}, dU), (hP, \bot_{\mathsf{c}}, uhP)\}$. *In order to decide whether* $\langle cDU, \mathsf{sc}, uhP\rangle$*, we check whether its* $\rho df_\bot$*-counterpart,* (cDU, sc, uhP)*, is derivable from the* $\rho df_\bot$*-counterpart of the portion of the graph we consider, that is,* $\{(cDU, \mathsf{sc}, hP), (cDU, \mathsf{sc}, dU), (hP, \bot_{\mathsf{c}}, uhP)\}$. *It is easy to check that there is no way of deriving* (cDU, sc, uhP) *from this graph.*

The semantics for defeasible $\rho df_\bot$ are defined with a ranking of $\rho df_\bot$-models: the lowest the rank of the model, the more expected the situation it describes is considered. As for the propositional and DL case [23], given a defeasible graph G its RC is determined by its *minimal* ranked model, that is, the model of G in which every $\rho df_\bot$-model is ranked as low as possible. The technical details can be found in [14, Sect. 3].

4. Conclusions

The main features of our approach are: (i) the defeasible $\rho df_\bot$ we propose remains syntactically a triple language by extending it with new predicate symbols with specific semantics; (ii) the logic is defined in such a way that any RDFS reasoner/store may handle the new predicates as ordinary terms if it does not want to take into account of the extra non-monotonic capabilities; (iii) the defeasible entailment decision procedure is built on top of the $\rho df_\bot$ entailment decision procedure, which in turn is an extension of the one for ρdf via some additional inference rules, favouring a potential implementation; (iv) the computational complexity of deciding entailment in ρdf and $\rho df_\bot$ are the same; and (v) defeasible entailment can be decided via a polynomial number of calls to an oracle deciding ground triple entailment in $\rho df_\bot$ and, in particular, deciding defeasible entailment can be done in polynomial time. While an extended version of the paper is under review at the moment, a technical report is online [14].

Acknowledgments

This research was partially supported by TAILOR (Foundations of Trustworthy AI – Integrating Reasoning, Learning and Optimization), a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

[1] P. A. Bonatti, M. Faella, I. Petrova, L. Sauro, A new semantics for overriding in description logics, Artificial Intelligence Journal 222 (2015) 1–48. URL: http://dx.doi.org/10.1016/j.artint.2014.12.010. doi:10.1016/j.artint.2014.12.010.

[2] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, H. Tompits, Combining answer set programming with description logics for the semantic web, Artificial Intelligence 172 (2008) 1495–1539. URL: http://dx.doi.org/10.1016/j.artint.2008.04.002. doi:10.1016/j.artint.2008.04.002.

[3] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, Well-founded semantics for description logic programs in the semantic web, ACM Transaction on Computational Logic 12 (2011) 11:1–11:41. URL: http://doi.acm.org/10.1145/1877714.1877717. doi:10.1145/1877714.1877717.

[4] L. Giordano, V. Gliozzi, N. Olivetti, G. L. Pozzato, A non-monotonic description logic for reasoning about typicality, Artificial Intelligence Journal 195 (2013) 165–202. URL: http://dx.doi.org/10.1016/j.artint.2012.10.004. doi:10.1016/j.artint.2012.10.004.

[5] B. Motik, R. Rosati, Reconciling description logics and rules, Journal of the ACM 57 (2010).

[6] K. Britz, G. Casini, T. Meyer, K. Moodley, U. Sattler, I. Varzinczak, Principles of klm-style defeasible description logics, ACM Trans. Comput. Logic 22 (2020). URL: https://doi.org/10.1145/3420258. doi:10.1145/3420258.

[7] A. Analyti, C. V. Damásio, G. Antoniou, Extended RDF: computability and complexity issues, Annals of Mathematics and Artificial Intelligence 75 (2015) 267–334. URL: https://doi.org/10.1007/s10472-015-9451-0. doi:10.1007/s10472-015-9451-0.

[8] G. Antoniou, Nonmonotonic rule systems on top of ontology layers, in: Proceedings of the 1st International Semantic Web Conference (ISWC-02), volume 2663 of *Lecture Notes in Computer Science*, Springer Verlag, Sardinia, Italia, 2002, pp. 394–398.

[9] T. Eiter, G. Ianni, R. Schindlauer, H. Tompits, Effective integration of declarative rules with external evaluations for semantic-web reasoning, in: The Semantic Web: Research and Applications, 3rd European Semantic Web Conference (ESWC-06), volume 4011 of *Lecture Notes in Computer Science*, Springer Verlag, 2006, pp. 273–287.

[10] G. Ianni, T. Krennwallner, A. Martello, A. Polleres, A rule system for querying persistent RDFS data, in: The Semantic Web: Research and Applications, 6th European Semantic Web Conference (ESWC-2009), 2009, pp. 857–862.

[11] D. Lehmann, M. Magidor, What does a conditional knowledge base entail?, Artif. Intell. 55 (1992) 1–60.

[12] S. Muñoz, J. Pérez, C. Gutierrez, Simple and Efficient Minimal RDFS, Web Semantics: Science, Services and Agents on the World Wide Web 7 (2009) 220–234. URL: http://dx.doi.org/10.1016/j.websem.2009.07.003. doi:10.1016/j.websem.2009.07.003.

[13] C. Gutiérrez, C. A. Hurtado, A. O. Mendelzon, J. Pérez, Foundations of semantic web databases, J. Comput. Syst. Sci. 77 (2011) 520–541. URL: https://doi.org/10.1016/j.jcss.2010.04.009. doi:10.1016/j.jcss.2010.04.009.

[14] G. Casini, U. Straccia, Defeasible RDFS via rational closure, CoRR abs/2007.07573 (2020). URL: https://arxiv.org/abs/2007.07573. arXiv:2007.07573.

[15] RDF, http://www.w3.org/RDF/, W3C, 2004.

[16] A. Hogan, M. Arenas, A. Mallea, A. Polleres, Everything you always wanted to know about blank nodes, J. Web Semant. 27-28 (2014) 42–69. URL: https://doi.org/10.1016/j.websem.2014.06.004. doi:10.1016/j.websem.2014.06.004.

[17] G. Casini, U. Straccia, T. Meyer, A polynomial time subsumption algorithm for nominal safe $\mathcal{ELO}_\bot$ under rational closure, Inf. Sci. 501 (2019) 588–620. URL: https://doi.org/10.1016/j.ins.2018.09.037. doi:10.1016/j.ins.2018.09.037.

[18] U. Straccia, G. Casini, A minimal deductive system for rdfs with negative statements, in: Principles of Knowledge Representation and Reasoning: Proceedings of the 19th International Conference, KR 2022, Haifa, Israel, July 31 - August 5, 2022, forth.

[19] S. Muñoz, J. Pérez, C. Gutiérrez, Minimal deductive systems for RDF, in: 4th European Semantic Web Conference (ESWC-07), volume 4519 of *Lecture Notes in Computer Science*, Springer Verlag, 2007, pp. 53–67.

[20] D. Lehmann, Another perspective on default reasoning, Annals of Mathematics and Artificial Intelligence 15 (1995) 61–82.

[21] G. Casini, U. Straccia, Defeasible inheritance-based description logics, Journal of Artificial Intelligence Research 48 (2013) 415–473.

[22] G. Casini, T. Meyer, K. Moodley, R. Nortjé, Relevant closure: A new form of defeasible reasoning for description logics, in: E. Fermé, J. Leite (Eds.), Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA-14), volume 8761 of *LNCS*, Springer, 2014, pp. 92–106.

[23] L. Giordano, V. Gliozzi, N. Olivetti, G. Pozzato, Semantic characterization of rational closure: From propositional logic to description logics, Artificial Intelligence Journal 226 (2015) 1–33.