Efficient TBox Reasoning with Value Restrictions Using the $\mathcal{FL}_o$*wer* Reasoner

Extended Abstract

Franz Baader, Patrick Koopmann, Friedrich Michel, Anni-Yasmin Turhan and Benjamin Zarriess

Institute of Theoretical Computer Science, Technische Universität Dresden, 01062 Dresden, Germany

Abstract

The inexpressive Description Logic (DL) $\mathcal{FL}_0$, which has conjunction and value restriction as its only concept constructors, had fallen into disrepute when it turned out that reasoning in $\mathcal{FL}_0$ w.r.t. general TBoxes is ExpTime-complete, that is, as hard as in the considerably more expressive logic $\mathcal{ALC}$. In the paper published in the journal Theory and Practice of Logic Programming, we rehabilitate $\mathcal{FL}_0$ by presenting a dedicated subsumption algorithm for $\mathcal{FL}_0$, which is much simpler than the tableau-based algorithms employed by highly optimized DL reasoners. Our experiments show that the performance of our novel algorithm, as prototypically implemented in our $\mathcal{FL}_o$*wer* reasoner, compares very well with that of the highly optimized reasoners. $\mathcal{FL}_o$*wer* can also deal with ontologies written in $\mathcal{FL}_\bot$, the extension of $\mathcal{FL}_0$ with the top and the bottom concept, by employing a polynomial-time reduction, shown in this paper, which eliminates the top and bottom concepts. We also investigate the complexity of reasoning in DLs related to the Horn-fragments of $\mathcal{FL}_0$ and $\mathcal{FL}_\bot$.

Keywords

Description Logic FL0, Value Restrictions, Efficient Reasoning, Horn-Fragments

Description Logics (DLs) [1, 2] are a well-investigated family of logic-based knowledge representation languages, which are frequently used to formalize ontologies for application domains such as the Semantic Web [3] or biology and medicine [4]. To define the important notions of such an application domain as formal concepts, DLs state necessary and sufficient conditions for an individual to belong to a concept. These conditions can be Boolean combinations of atomic properties required for the individual (expressed by concept names) or properties that refer to relationships with other individuals and their properties (expressed as role restrictions). For example, the concept of a parent that has only daughters can be formalized by the concept description $C := \exists \mathit{child}.\top \sqcap \forall \mathit{child}.(\mathit{Female} \sqcap \mathit{Human})$, which uses the concept names *Female* and *Human* and the role name *child* as well as the concept constructors top concept ($\top$), conjunction ($\sqcap$), existential restriction ($\exists r.D$), and value restriction ($\forall r.D$). The concept description $\mathit{Human} \sqcap \forall \mathit{child}.\bot$, which additionally uses the concept constructor bottom concept ($\bot$), formalizes the concept of humans without children. Constraints on the interpretation of concept and role names can be formulated as general concept inclusions (GCIs). For example,

DL 2022: 35th International Workshop on Description Logics, August 7–10, 2022, Haifa, Israel

franz.baader@tu-dresden.de (F. Baader); patrick.koopmann@tu-dresden.de (P. Koopmann); friedrich.michel@tu-dresden.de (F. Michel); anni-yasmin.turhan@tu-dresden.de (A. Turhan); benjamin.zarriess@tu-dresden.de (B. Zarriess)

0000-0002-4049-221X (F. Baader); 0000-0001-5999-2583 (P. Koopmann); 0000-0001-6336-335X (A. Turhan)

the GCIs $Human \sqsubseteq \forall child.Human$ and $\exists child.Human \sqsubseteq Human$ say that humans have only human children, and that they are the only ones that can have human children. DL systems provide their users with reasoning services that allow them to derive implicit knowledge from the explicitly stated one. In our example, the above GCIs imply that elements of our concept C also belong to the concept $D := Human \sqcap \exists child.Human$, i.e., C is subsumed by D w.r.t. these GCIs. A specific DL is determined by which kind of concept constructors are available.

In the early days of DL research, the inexpressive DL $\mathcal{FL}_0$, which has only conjunction and value restriction as concept constructors, was considered to be the smallest possible DL. In fact, when providing a formal semantics for so-called property edges of semantic networks in the first DL system KL-ONE [5], value restrictions were used. For this reason, the language for constructing concepts in KL-ONE and all of the other early DL systems [6, 7, 8, 9] contained $\mathcal{FL}_0$. It came as a surprise when it was shown that subsumption reasoning w.r.t. acyclic $\mathcal{FL}_0$ TBoxes (a restricted form of GCIs) is co-NP-hard [10]. The complexity increases when more expressive forms of TBoxes are used: for cyclic TBoxes to PSpace [11, 12] and for general TBoxes consisting of GCIs even to ExpTime [13, 14]. Thus, w.r.t. general TBoxes, subsumption reasoning in $\mathcal{FL}_0$ is as hard as subsumption reasoning in $\mathcal{ALC}$, its closure under negation [15].

These negative complexity results for $\mathcal{FL}_0$ were one of the reasons why the attention in the research of inexpressive DLs shifted from $\mathcal{FL}_0$ to $\mathcal{EL}$, which is obtained from $\mathcal{FL}_0$ by replacing value restriction with existential restriction as a constructor. In fact, subsumption reasoning in $\mathcal{EL}$ stays polynomial even in the presence of general TBoxes [16]. The reasoning method employed in [16], which is nowadays called consequence-based reasoning, can be used to establish a PTime complexity upper bound also for reasoning in the extension $\mathcal{EL}^+$ of $\mathcal{EL}$ [13]. This approach also applies to Horn fragments of expressive DLs such as $\mathcal{SHIQ}$, for which reasoning is ExpTime-complete, but consequence-based reasoning approaches behave considerably better in practice than the usual tableau-based approaches for expressive DLs [17].

The DL $\mathcal{FL}_0$ is not Horn,[1] but it shares with $\mathcal{EL}^+$ and Horn-$\mathcal{SHIQ}$ that (general) TBoxes have canonical models, i.e., models such that a subsumption relationship between concept names follows from the TBox if and only if it holds in the canonical model. Consequence-based reasoning basically generates these models. However, whereas the canonical models for $\mathcal{EL}$ and Horn-$\mathcal{SHIQ}$ are respectively of polynomial and exponential size, the canonical models for $\mathcal{FL}_0$, called least functional models in [20], may be infinite. These least functional models can, however, be represented using so-called looping tree automata (LTAs). This fact is used in [20] to reduce the subsumption problem in $\mathcal{FL}_0$ to the emptiness problem for LTAs. While this automata-based subsumption algorithm is worst-case optimal (i.e., it runs in exponential time), it has the disadvantage that it always needs exponential time, since the first step is to build an exponentially large LTA. An alternative approach, which constructs a finite part of the least functional model, was described in [21], where also experimental results for a first prototypical implementation were reported.

In the paper [22], on which this extended abstract reports, we build on and extend the results from [21]. We devise a novel algorithm for deciding subsumption w.r.t. general $\mathcal{FL}_0$ TBoxes, describe an implementation of it in the new $\mathcal{FL}_o$*wer* reasoner,[2] and report on an evaluation

[1] Actually, reasoning in its Horn fragment is PTime [18, 19].
[2] https://github.com/attalos/fl0wer

of $\mathcal{FL}_o$*wer* on a large collection of ontologies, which shows that $\mathcal{FL}_o$*wer* competes well with existing highly optimized DL reasoners. Basically, the algorithm generates "large enough" parts of the least functional model and achieves termination using a blocking mechanism similar to the ones employed by tableau-based reasoners. However, unlike tableaux algorithms for expressive DLs, this algorithm is deterministic. The key idea of the implementation is to apply the TBox statements like rules and to use a variant of the well-known Rete algorithm for rule application [23], adapted to the case without negation. To create a large set of challenging $\mathcal{FL}_0$ ontologies we have used, on the one hand, the OWL 2 EL ontologies of the OWL reasoner competition [24], transformed into $\mathcal{FL}_0$ by replacing existential restrictions with value restrictions and omitting too small ontologies as too easy. On the other hand, we have extracted $\mathcal{FL}_0$ sub-ontologies of decent size from the ontologies of the Manchester OWL Corpus (MOWLCorp).[3]

After introducing $\mathcal{FL}_0$ and its extension $\mathcal{FL}_\bot$ with the top ($\top$) and the bottom ($\bot$) concepts, the paper recalls the characterization of subsumption based on least functional models from [20]. It introduces a normal form for $\mathcal{FL}_0$ TBoxes, and then shows that the bottom concept $\bot$ and the top concept $\top$ can be simulated by such TBoxes. Then the new subsumption algorithm for $\mathcal{FL}_0$ is introduced and proved to be sound, complete, and terminating. Subsequently, Horn fragments of $\mathcal{FL}_0$ and $\mathcal{FL}_\bot$ are investigated which is an extension compared to [21]. First, it is shown that, for Horn-$\mathcal{FL}_0$, our algorithm can be restricted such that it runs in polynomial time. A polynomial upper bound for subsumption in Horn-$\mathcal{FL}_0$ has already been shown in [18, 19] for an extension of Horn-$\mathcal{FL}_0$ that contains $\bot$. However, this extension is weaker than Horn-$\mathcal{FL}_\bot$. In fact, we also show in this paper that subsumption in Horn-$\mathcal{FL}_\bot$ is PSpace-complete, and that it becomes ExpTime-complete in a small extension of Horn-$\mathcal{FL}_\bot$. After describing how to realize the novel algorithm based on Rete and how to optimize the algorithm from [21], the paper presents experimental results on larger data sets than in [21]. For these it evaluates several optimizations of the algorithm, and compares its performance with that of existing highly optimized DL reasoners.

Summing up, the main contribution of the paper is a novel algorithm for deciding subsumption in the DL $\mathcal{FL}_0$ w.r.t. general TBoxes, and a practical demonstration that this algorithm is easy to implement and behaves surprisingly well on large ontologies. Our reasoner $\mathcal{FL}_o$*wer* outperforms state-of-the art DL reasoners for testing subsumption and for classifying general TBoxes. One may ask, however, why a dedicated reasoner for $\mathcal{FL}_0$ is needed, given the facts that the worst-case complexity of reasoning in $\mathcal{FL}_0$ is as high as for the considerably more expressive DL $\mathcal{ALC}$ and that there are very few pure $\mathcal{FL}_0$ ontologies available. We argue that such a dedicated reasoner may turn out to be very useful. First, the latter fact could be due to a chicken and egg problem: as long as no dedicated reasoner for $\mathcal{FL}_0$ is available, there is no incentive to restrict the expressiveness to $\mathcal{FL}_0$ when creating an ontology. When extracting our test ontologies, we observed that quite a number of application ontologies have large $\mathcal{FL}_0$ fragments. Second, regarding the former fact, it is well-known in the DL community that worst-case complexity results are not always a good indication for how hard reasoning turns out to be in practice. Third, some DL reasoners such as Konclude[4] and MORe [25] make use of specialized algorithms for certain language fragments as part of their overall reasoning approach, with impressive

[3] https://zenodo.org/record/16708
[4] konclude.com

improvements of the performance. Our efficient subsumption algorithm for $\mathcal{FL}_0$ may turn out to be useful in this context. Finally, quite a number of non-standard reasoning tasks in $\mathcal{FL}_0$ w.r.t. general TBoxes have recently been investigated [26, 27, 20, 28]. The algorithms developed for solving these tasks usually depend on sub-procedures that perform subsumption tests or that use the least functional model directly. Our reasoner $\mathcal{FL}_o$*wer* thus provides us with an efficient base for implementing such non-standard inferences.

Acknowledgments

This work was partially supported by the AI competence center ScaDS.AI Dresden/Leipzig and the Deutsche Forschungsgemeinschaft (DFG), Grant 389792660 within TRR 248.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.

[2] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[3] I. Horrocks, P. F. Patel-Schneider, F. van Harmelen, From SHIQ and RDF to OWL: The making of a web ontology language, Journal of Web Semantics 1 (2003) 7–26.

[4] R. Hoehndorf, P. N. Schofield, G. V. Gkoutos, The role of ontologies in biological and biomedical research: A functional perspective, Brief. Bioinform. 16 (2015) 1069–1080.

[5] R. J. Brachman, J. G. Schmolze, An overview of the KL-ONE knowledge representation system, Cognitive Science 9 (1985) 171–216.

[6] R. J. Brachman, D. L. McGuinness, P. F. Patel-Schneider, L. Alperin Resnick, A. Borgida, Living with CLASSIC: When and how to use a KL-ONE-like language, in: J. F. Sowa (Ed.), Principles of Semantic Networks, Morgan Kaufmann, Los Altos, 1991, pp. 401–456.

[7] C. Peltason, The BACK system — an overview, SIGART Bull. 2 (1991) 114–119.

[8] E. Mays, R. Dionne, R. Weida, K-REP system overview, SIGART Bull. 2 (1991).

[9] W. A. Woods, J. G. Schmolze, The KL-ONE family, in: F. W. Lehmann (Ed.), Semantic Networks in Artificial Intelligence, Pergamon Press, 1992, pp. 133–178. Published as a special issue of *Computers & Mathematics with Applications*, Volume 23, Number 2–9.

[10] B. Nebel, Terminological reasoning is inherently intractable, Artificial Intelligence 43 (1990) 235–249.

[11] F. Baader, Terminological cycles in KL-ONE-based knowledge representation languages, in: Proc. of the 8th Nat. Conf. on Artificial Intelligence (AAAI'90), Boston (Ma, USA), 1990, pp. 621–626.

[12] Y. Kazakov, H. de Nivelle, Subsumption of concepts in $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete, in: Proc. of the 2003 Description Logic Workshop (DL 2003), CEUR Electronic Workshop Proceedings, http://CEUR-WS.org/Vol-81/, 2003.

[13] F. Baader, S. Brandt, C. Lutz, Pushing the $\mathcal{EL}$ envelope, in: L. P. Kaelbling, A. Saffiotti

(Eds.), Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005), Morgan Kaufmann, Los Altos, Edinburgh (UK), 2005, pp. 364–369.

[14] M. Hofmann, Proof-theoretic approach to description-logic, in: P. Panangaden (Ed.), Proc. of the 20th IEEE Symp. on Logic in Computer Science (LICS 2005), IEEE Computer Society Press, 2005, pp. 229–237.

[15] K. Schild, A correspondence theory for terminological logics: Preliminary report, in: Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91), 1991, pp. 466–471.

[16] S. Brandt, Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else?, in: R. L. de Mántaras, L. Saitta (Eds.), Proc. of the 16th Eur. Conf. on Artificial Intelligence (ECAI 2004), 2004, pp. 298–302.

[17] Y. Kazakov, Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies, in: C. Boutilier (Ed.), Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009), IJCAI/AAAI, 2009, pp. 2040–2045.

[18] M. Krötzsch, S. Rudolph, P. Hitzler, Complexity boundaries for Horn description logics, in: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada, AAAI Press, 2007, pp. 452–457. URL: http://www.aaai.org/Library/AAAI/2007/aaai07-071.php.

[19] M. Krötzsch, S. Rudolph, P. Hitzler, Complexities of Horn description logics, ACM Trans. Comput. Log. 14 (2013) 2:1–2:36. URL: https://doi.org/10.1145/2422085.2422087. doi:10.1145/2422085.2422087.

[20] F. Baader, O. Fernandez Gil, M. Pensel, Standard and non-standard inferences in the description logic $\mathcal{FL}_0$ using tree automata, in: D. D. Lee, A. Steen, T. Walsh (Eds.), GCAI-2018, 4th Global Conference on Artificial Intelligence, volume 55 of *EPiC Series in Computing*, EasyChair, 2018, pp. 1–14. URL: http://www.easychair.org/publications/paper/H6d9.

[21] F. Michel, A.-Y. Turhan, B. Zarrieß, Efficient TBox reasoning with value restrictions—introducing the $\mathcal{FL}_o$*wer* reasoner, in: P. Fodor, M. Montali (Eds.), Proceedings of the 3rd International Joint Conference on Rules and Reasoning (RuleML+RR 2019), LNCS, Springer, Bolzano, Italy, 2019. doi:https://dx.doi.org/10.1007/978-3-030-31095-0_9.

[22] F. Baader, P. Koopmann, F. Michel, A.-Y. Turhan, B. Zarriess, Efficient TBox reasoning with value restrictions using the $\mathcal{FL}_0$wer reasoner, Theory and Practice of Logic Programming 22 (2022) 162–192. doi:10.1017/S1471068421000466.

[23] C. Forgy, Rete: A fast algorithm for the many patterns/many objects match problem, Artif. Intell. 19 (1982) 17–37. URL: https://doi.org/10.1016/0004-3702(82)90020-0. doi:10.1016/0004-3702(82)90020-0.

[24] B. Parsia, N. Matentzoglu, R. S. Gonçalves, B. Glimm, A. Steigmiller, The owl reasoner evaluation (ore) 2015 competition report, Journal of Automated Reasoning 59 (2017) 455–482.

[25] A. A. Romero, B. Cuenca Grau, I. Horrocks, More: Modular combination of OWL reasoners for ontology classification, in: International Semantic Web Conference (1), volume 7649 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 1–16.

[26] F. Baader, P. Marantidis, A. Okhotin, Approximate unification in the description logic $\mathcal{FL}_0$, in: L. Michael, A. C. Kakas (Eds.), Logics in Artificial Intelligence - 15th European Conference, JELIA 2016, Proceedings, volume 10021 of *Lecture Notes in Computer Sci-*

ence, 2016, pp. 49–63. URL: https://doi.org/10.1007/978-3-319-48758-8_4. doi:10.1007/978-3-319-48758-8_4.
[27] F. Baader, O. Fernandez Gil, P. Marantidis, Matching in the description logic $\mathcal{FL}_0$ with respect to general TBoxes, in: G. Barthe, G. Sutcliffe, M. Veanes (Eds.), LPAR-22. 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, volume 57 of *EPiC Series in Computing*, EasyChair, 2018, pp. 76–94. URL: http://www.easychair.org/publications/paper/XrXz.
[28] F. Baader, P. Marantidis, M. Pensel, The data complexity of answering instance queries in $\mathcal{FL}_0$, in: P. Champin, F. L. Gandon, M. Lalmas, P. G. Ipeirotis (Eds.), Companion of the The Web Conference WWW, ACM, 2018, pp. 1603–1607. URL: https://doi.org/10.1145/3184558.3191618. doi:10.1145/3184558.3191618.