From Tableaux to Automata for Description Logics

Franz Baader, Jan Hladik, and Carsten Lutz
Theoretical Computer Science, TU Dresden,
D-01062 Dresden, Germany
{baader,hladik,lutz}@tcs.inf.tu-dresden.de

Frank Wolter
Department of Computer Science, University of Liverpool
Liverpool L69 7ZF, U.K.
frank@csc.liv.ac.uk

Description logics (DLs) [1] are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. The name *description logics* is motivated by the fact that, on the one hand, the important notions of the domain are described by *concept descriptions*, i.e., expressions that are built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the concept and role constructors provided by the particular DL. On the other hand, DLs differ from their predecessors, such as semantic networks and frames [22, 18], in that they are equipped with a formal, *logic*-based semantics, which can, e.g., be given by a translation into first-order predicate logic.

Knowledge representation systems based on description logics (DL systems) [24, 19] provide their users with various inference capabilities (like subsumption and instance checking) that allow them to deduce implicit knowledge from the explicitly represented knowledge. In order to ensure a reasonable and predictable behavior of a DL system, these inference problems should at least be decidable, and preferably of low complexity. Consequently, the expressive power of the DL in question must be restricted in an appropriate way. If the imposed restrictions are too severe, however, then the important notions of the application domain can no longer be expressed. Investigating this trade-off between the expressivity of DLs and the complexity of their inference problems has been one of the most important issues in DL research (see [8] for an overview of complexity results).

The focus of this research has, however, changed in the last 15 years. In the beginning of the 1990ies, DL researchers investigated the border between tractable and intractable DLs [10, 11], and systems that employed so-called structural subsumption algorithms, which first normalize the concept descriptions, and then recursively compare the syntactic structure of the normalized descriptions, were still prevalent [21, 16, 17, 20]. It quickly turned out, however, that structural subsumption algorithms can handle only very inexpressive languages, and that one cannot expect a DL of reasonable expressive power to have tractable inference problems. For expressive DLs, tableau-based inference procedures turned out to be quite useful. After the first such tableau-based subsumption algorithm was developed by Schmidt-Schauß and Smolka [23] for the DL $\mathcal{ALC}$, this approach was extended to various other DLs and also to other inference problems such as the instance problem (see [5] for an overview).

Most of these early tableau-based algorithms for DLs were of optimal worst-case complexity: they treated DLs with a PSPACE-complete subsumption problem, and the algorithms needed only polynomial space. Thus, by designing a tableau-based algorithm for such a DL one could solve two problems simultaneously: prove an optimal complexity upper-bound, and describe an algorithm that is easy to implement and optimize [2, 14], thus yielding a practical reasoning system for this DL. Modern tableau-based DL reasoners such as FaCT [13] and RACER [12] are based on very expressive DLs (like $\mathcal{SHIQ}$ [15]), which have an ExpTime-complete subsumption problem. Despite the high worst-case complexity of the underlying logics, the systems FaCT and RACER behave quite well in realistic applications. This is mainly due to the fact that their implementors have developed a great variety of sophisticated optimization techniques for tableau-based algorithms (see [14] for an overview of these techniques). Tableau-based algorithms are, however, notoriously bad at proving ExpTime upper-bounds.[1] In many cases, ExpTime upper-bounds are easily established using automata-based approaches (see, e.g., Section 5.3 in [7]). However, to the best of our knowledge, there exist no practical DL reasoners based on automata techniques. Until now, it was thus common practice to devise two different algorithms for every ExpTime-complete DL, an automata-based one for establishing the exact worst-case complexity, and a tableau-based one for the implementation.

This talk investigates the (rather close) relationship between automata- and tableau-based algorithms. To be more precise, we introduce an abstract notion of what a tableau-based algorithm is, and then show, on this abstract level, how tableau-based algorithms can be converted into automata-based algorithms. In particular, this allows us to characterize a large class of tableau-based algorithms that imply an ExpTime upper-bound for reasoning in the DLs for which such an algorithm exists. We consider this to be a very useful result since, in many cases, it eliminates the need for developing two algorithms for the same DL: one can now design a tableau-based algorithm,

[1] The only such result we know of [9] treats the case of $\mathcal{ALC}$ with general concept inclusions (GCIs), and even in this simple case the algorithm is very complicated.

use our general result to obtain an ExpTime upper-bound, and then base a practical implementation on the very same algorithm.

This work has been published in Fundamenta Informaticae [3] and in a preliminary version in the proceedings of LPAR 2003 [4].

References

[1] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[2] Franz Baader, Enrico Franconi, Bernhard Hollunder, Bernhard Nebel, and Hans-Jürgen Profitlich. An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. *Applied Artificial Intelligence. Special Issue on Knowledge Base Management*, 4:109–132, 1994.

[3] Franz Baader, Jan Hladik, Carsten Lutz, and Frank Wolter. From tableaux to automata for description logics. *Fundamenta Informaticae*, 57(2–4):247–279, 2003.

[4] Franz Baader, Jan Hladik, Carsten Lutz, and Frank Wolter. From tableaux to automata for description logics. In Moshe Vardi and Andrei Voronkov, editors, *Proceedings of the 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2003)*, volume 2850 of *Lecture Notes in Computer Science*, pages 1–32, Almaty, Kazakhstan, 2003. Springer-Verlag.

[5] Franz Baader and Ulrike Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.

[6] Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation.* Morgan Kaufmann, Los Altos, 1985.

[7] Diego Calvanese and Giuseppe DeGiacomo. Expressive description logics. In *[1]*, pages 178–218. 2003.

[8] Francesco Donini. Complexity of reasoning. In *[1]*, pages 96–136. 2003.

[9] Francesco Donini and Fabio Massacci. EXPTIME tableaux for $\mathcal{ALC}$. *Acta Informatica*, 124(1):87–138, 2000.

[10] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proc. of the 2nd Int. Conf. on the*

Principles of Knowledge Representation and Reasoning (KR'91), pages 151–162. Morgan Kaufmann, Los Altos, 1991.

[11] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. Tractable concept languages. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 458–463, Sydney (Australia), 1991.

[12] Volker Haarslev and Ralf Möller. RACER system description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001)*, 2001.

[13] Ian Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.

[14] Ian Horrocks. Implementation and optimization techniques. In *[1]*, pages 306–346. 2003.

[15] Ian Horrocks, Ulrike Sattler, and Stefan Tobies. Practical reasoning for very expressive description logics. *J. of the Interest Group in Pure and Applied Logic*, 8(3):239–264, 2000.

[16] Robert MacGregor. The evolving technology of classification-based knowledge representation systems. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 385–400. Morgan Kaufmann, Los Altos, 1991.

[17] E. Mays, R. Dionne, and R. Weida. K-REP system overview. *SIGART Bull.*, 2(3), 1991.

[18] Marvin Minsky. A framework for representing knowledge. In J. Haugeland, editor, *Mind Design*. The MIT Press, 1981. A longer version appeared in *The Psychology of Computer Vision* (1975). Republished in [6].

[19] Ralf Möller and Volker Haarslev. Description logic systems. In *[1]*, pages 282–305. 2003.

[20] Peter F. Patel-Schneider, Deborah L. McGuiness, Ronald J. Brachman, Lori Alperin Resnick, and Alexander Borgida. The CLASSIC knowledge representation system: Guiding principles and implementation rational. *SIGART Bull.*, 2(3):108–113, 1991.

[21] C. Peltason. The BACK system — an overview. *SIGART Bull.*, 2(3):114–119, 1991.

[22] M. Ross Quillian. Word concepts: A theory and simulation of some basic capabilities. *Behavioral Science*, 12:410–430, 1967. Republished in [6].

[23] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[24] William A. Woods and James G. Schmolze. The KL-ONE family. In F. W. Lehmann, editor, *Semantic Networks in Artificial Intelligence*, pages 133–178. Pergamon Press, 1992. Published as a special issue of *Computers & Mathematics with Applications*, Volume 23, Number 2–9.