The Use of Description Logics as Database Query Languages

Klaus Schild
Daimler-Benz AG
Research and Technology
Knowledge Based Systems
Alt-Moabit 96a, D-10559 Berlin, Germany
e-mail: schild@DBresearch-berlin.de

Description Logics are knowledge representation languages set up by the development of the Kl-One system [1]. They are used to capture the taxonomy of an application domain and to describe the application domain itself in terms of this taxonomy. These specific logics employ user-friendly variable-free notations. One of their major characteristics is their clear semantics. Without such a formal semantics, it would be impossible to state what exactly is represented by a particular representation. In other words, without formal semantics, representations would have no meaning outside the particular system in which they reside—preventing the knowledge fixed in the representation from being re-used.

The line of research set up by the Kl-One project can be called successful in the long run. An indication for its success certainly is that the most recent successor of Kl-One, AT&T's Classic system, eventually reached the realm of a large-scale industrial application [6]. This success, however, should not obscure the fact that there is a fundamental dilemma from which all description logics suffer. In fact, despite their limited expressive power, basic inferential services such as classifying new terms into a taxonomy cannot be implemented efficiently. In particular, it is known that even in the very smallest description logic's setting, basic inferences are co-NP-hard [2].

In [4] we have shown that this fundamental dilemma can in principle be circumvented. In particular, we were able to demonstrate that tractability can generally be obtained just by eliminating any *incompleteness* from a knowledge base while the taxonomy is left unchanged. This remains true even for the most powerful description logic ever considered. The description logic we have paid attention to can be called with full right *universal* in that it encompasses all language repositories known from traditional description logics. This enables the universal description logic to define many standard data structures such as trees or directed acyclic graphs in an elegant way. In addition to traditional constructs, the universal description logic includes a general means of recursion. As is not unusual in computer science, we handled recursion with the help of least and greatest fixed-point operators. The technique employed is actually a generalization of the technique presented in [3]. These fixed-point operators turned out to be indispensable as soon as more involved concepts such as balanced trees are to be modeled. As it stands, this tractability result is of great importance. Actually, it is the very first tractability result established for a description logic which takes taxonomies into account.

On the other hand, our result can be viewed as building a bridge between traditional knowledge representation and databases. As a matter of fact, our tractability result heavily depends on the presupposition that any incomplete knowledge can be eliminated from a knowledge base. The ability to express incomplete knowledge is, of course, the very characteristic separating knowledge representation from databases. A knowledge base which is complete in this sense is, in fact, nothing but a relational database. Consequently, when viewed from the database point of view, our tractability result demonstrates that a universal description logic can be used as a powerful but tractable query language for relational databases. In this connection, it is important to note that our tractability result is to be understood in the sense of [5] in terms of the combined complexity rather than the far weaker notion of data complexity. Of course, this database point of view on description logics gives rise to several questions hardly investigated up till now. These include questions of the following kind.

1. We have shown that a universal description logic can serve as a tractable query language for databases. This means that queries to databases phrased in this description logic can be evaluated in polynomial time. But is it also the case that the universal description logic covers *all* polynomial queries?
2. How does the query power of the universal description logic relate to other more traditional database query languages?
3. Is it possible to extend our tractability result to deal with essential additional features common in relational databases such as null values?

We discussed all these questions at the workshop. A thorough investigation can be found in [4].

References

[1] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[2] B. Nebel. Terminological Reasoning is Inherently Intractable. *Artificial Intelligence*, 43:235–249, 1990.

[3] K. Schild. Terminological cycles and the propositional μ-calculus. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 509–520, Bonn, FRG, 1994.

[4] K. Schild. *A Tractable Query Language for Knowledge and Data Bases Based on a Universal Description Logic and Logics of Programs.* PhD thesis, German Research Center for Artificial Intelligence–DFKI GmbH, Saarbrücken, FRG, 1995. Forthcoming.

[5] M. Y. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 137–146, San Francisco, CA, 1982.

[6] J. R. Wright, E. S. Weixelbaum, G. T. Vesonder, K. Brown, S. R. Palmer, J. I. Berman, and H. H. Moore. A knowledge-based configurator that supports sales, engineering, and manufacturing at AT&T network systems. *AI-Magazine*, 14(3):69–80, 1993.