Automated reasoning with nonmonotonic logics (INVITED TALK)

Mirosław Truszczyński

Department of Computer Science
University of Kentucky
Lexington, KY 40506-0027, USA
{mirek}@ms.uky.edu

The field of nonmonotonic logics was originated about twenty years ago in an effort to build efficient knowledge representation formalisms [Rei78, Rei80, McC80, MD80, Moo84]. Since then solid theoretical foundations of nonmonotonic logics have been established [MT93, Bre91, Bes89, Rei87]. It was expected that nonmonotonic logics would be able to model formal aspects of commonsense reasoning and that their computational properties would be better than those of classical logic. Computational complexity results obtained in recent years [MT91, Got92, EG92, Nie92, EG93] were, however, discouraging. Decision problems associated with nonmonotonic reasoning, even when restricted to the propositional case, are usually complete for classes on the first and second levels of polynomial hierarchy.

Are, then, nonmonotonic logics a viable approach to knowledge representation? The complexity theory, concerned with the worst case behavior, does not tell the whole story. Nonmonotonic logics seem to allow for more concise representations of domain knowledge [CDS94, CDS95, GKPS95]. And, ultimately, practicality of nonmonotonic reasoning systems can only be decided experimentally.

In this talk, we will describe basic algorithmic techniques used in building automated reasoning systems for nonmonotonic logics. We will give an overview of the complexity results, including recent results on conciseness of representations by nonmonotonic systems. We will then describe several current implementations of nonmonotonic logics including the systems by Niemelä and Simons, by Warren and its collaborators and the system DeReS developed by our group in the University of Kentucky [1]. We will discuss implications this work may have for the field of combinatorial optimization and, more generally, for the constraint satisfaction problem.

Our main goal in the talk will be to identify and discuss major challenges for automated reasoning systems based on nonmonotonic logics.

First, nonmonotonic logics have an inherent second order flavor. A logic program can be viewed as an encoding of the *family* of its stable models and a default

[1] Systems by Niemelä and Simons, and by Warren and his group are discussed in more detail elsewhere in the Proceedings. DeReS is described in [CMT96]. It can be obtained by anonymous ftp from *al.cs.engr.uky.edu* by retrieving the file *DeReS.tar.gz* from the directory *cs/software/logic*.

theory represents the *family* of its extensions. This distinguishes nonmonotonic systems from standard first-order logic based systems such as PROLOG, where a program serves as a computational mechanism for finding all *elements* of some Herbrand universe (or, equivalently, *one* subset of the Herbrand base). This indicates that nonmonotonic logics can be used when we are interested in one or more *subsets* of some universe and not necessarily in individual elements of this universe.

This general observation and its implications received relatively little attention so far. This needs to change as the observation has important implications. It seems to indicate that intended applications of nonmonotonic programming systems may be different from those commonly associated with automated reasoning. It also implies that the standard, PROLOG-directed, methodology of writing logic programs may not apply to systems based on nonmonotonic logics. New techniques are needed that will allow us represent and encode basic operations on families of sets rather than on sets. In a word, we need a methodology of programming with nonmonotonic systems.

Recent years resulted in several implementations of reasoning systems based on nonmonotonic logics. What did we learn from this effort? Three or four years ago, it was a commonly accepted opinion in the community that the field of nonmonotonic reasoning is now stagnating. The hope was that building working systems would provide a vindication to our theoretical work. My contention is that this expectation has not been met yet. In my view, there are two reasons. Current implementations are really only the first generation of programming environments based on nonmonotonic logics (versions 0.x, if you will). None of the existing systems have been thoroughly tested, there are still general lessens to be learned from experimentations, and not all of those few ones that were learned found their way yet into revised versions of systems we are building. At the very heart of the problem is lack of test data that supports comparative studies as well as allows testing scalability of methods with growing problem sizes.

Our work, in the University of Kentucky, on TheoryBase is an effort to provide some solution to this problem. Theorybase[2] generates test default theories and logic programs. These theories and programs are specified by several parameters which give the user control over their sizes and other properties such as existence or nonexistence of extensions and stable models, stratifiability, etc.

While useful, TheoryBase is not an ultimate answer to the problems of testing and benchmarking data. The second, and more critical, reason why implementing nonmonotonic reasoning systems has not yet opened new and exciting avenues for the LPNMR community is the lack of real application domains. The problem has been recognized for quite some time. But now we do not have any excuses to avoid it any more. When we develop implementations, why are we doing this? Who, in our expectations, will be using these systems and for what purpose?

[2] TheoryBase is described in [CMMT95]. It can be obtained by anonymous ftp from *al.cs.engr.uky.edu* by retrieving file *TheoryBase.tar.gz* from the directory *cs/software/logic*.

In what applications the technology we are developing has a chance to become commonly accepted and used? I will not offer any answers to these questions. But the time has come to seriously ponder upon them.

References

[Bes89] P. Besnard. *An introduction to default logic.* Springer-Verlag, Berlin, 1989.

[Bre91] G. Brewka. *Nonmonotonic reasoning: logical foundations of commonsense.* Cambridge University Press, Cambridge, UK, 1991.

[CDS94] M. Cadoli, F.M. Donini, and M. Schaerf. Is intractability of non-monotonic reasoning a real drawback? In *Proceedings of AAAI-94*, pages 946–951, Seattle, USA, 1994.

[CDS95] M. Cadoli, F.M. Donini, and M. Schaerf. On compact representations of propositional circumscription. In *Proceedings of STACS-95*, 1995.

[CMMT95] P. Cholewiński, W. Marek, A. Mikitiuk, and M. Truszczyński. Experimenting with nonmonotonic reasoning. In *Proceedings of the ICLP-95*, MIT Press, 1995,

[CMT96] P. Cholewiński, W. Marek, and M. Truszczyński. Default reasoning system DeReS. In *Proceedings of KR-96*. Morgan-Kaufmann, 1996.

[EG92] T. Eiter and G. Gottlob. Complexity of reasoning with parsimonious and moderately grounded expansions. *Fundamenta Inf.*, 17:31–53, 1992.

[EG93] T. Eiter and G. Gottlob. Propositional circumscription and extended closed world reasoning are Π_2^P-complete. *Theoretical Computer Science*, 114:231 – 245, 1993.

[GKPS95] G. Gogic, H. Kautz, Ch. Papadimitriou, and B. Selman. Compactness of knowledge representation: A comparative analysis. In *Proceedings of IJCAI-95*, pages 862–869. Morgan Kaufmann, 1995.

[Got92] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2:397–425, 1992.

[McC80] J. McCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[MD80] D. McDermott and J. Doyle. Nonmonotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[Moo84] R.C. Moore. Possible-world semantics for autoepistemic logic. In R. Reiter, editor, *Proceedings of the workshop on non-monotonic reasoning*, pages 344–354, 1984.

[MT91] W. Marek and M. Truszczyński. Autoepistemic logic. *Journal of the ACM*, 38:588–619, 1991.

[MT93] W. Marek and M. Truszczyński. *Nonmonotonic logics; context-dependent reasoning.* Berlin: Springer-Verlag, 1993.

[Nie92] I. Niemelä. On the decidability and complexity of autoepistemic reasoning. *Fundamenta Inf.*, 17:117–155, 1992.

[Rei78] R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and data bases*, pages 55–76. Plenum Press, 1978.

[Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[Rei87] R. Reiter. Nonmonotonic reasoning. *Ann. Rev. Comput. Sci.*, 2:147–186, 1987.

This article was processed using the LaTeX macro package with LLNCS style