Applications of Transaction Logic to Knowledge Representation

Anthony J. Bonner*
University of Toronto
Department of Computer Science
Toronto, Ontario M5S 1A4, Canada
bonner@db.toronto.edu

Michael Kifer†
SUNY at Stony Brook
Department of Computer Science
Stony Brook, NY 11790, U.S.A.
kifer@cs.sunysb.edu

Abstract

We present applications of the recently proposed *Transaction Logic*—an extension of classical logic that accounts in a clean and declarative fashion for the phenomenon of state changes in knowledge bases, including logic programs and databases. Transaction Logic has a natural model theory and a sound-and-complete proof theory, but unlike many other logics, it allows users to *program* transactions. Its semantics also leads naturally to features whose amalgamation in a single logic has proved elusive in the past. These features include both hypothetical *and* committed updates, static and dynamic constraints, nondeterministic actions, nested transactions, bulk updates, view updates, active databases, subjunctive queries, and more. Finally, Transaction Logic holds promise as a logical model of hitherto non-logical phenomena, including *procedural knowledge* in AI, and the *behavioral* aspect of object-oriented databases, especially methods with side effects. This paper outlines the model theory of Transaction Logic, and then focuses on some of its applications to AI, including action definition and execution, planning, and dynamic constraints.

This paper appears in *Proceedings of the International Conference on Temporal Logic (ICTL)*, pages 67–81. Springer Verlag, Lecture Notes in Artificial Intelligence, volume 827. Conference held in Bonn, Germany, July 11-14 1994.

This paper is available at the following URL:
`ftp://db.toronto.edu/pub/bonner/papers/transaction.logic/ictl94.ps.gz`

*Work supported in part by an Operating Grant from the Natural Sciences and Engineering Research Council of Canada and by a Connaught Grant from the University of Toronto.

†Supported in part by NSF grant CCR-9102159. Part of this work is done during sabbatical year at the University of Toronto. Support of Computer Systems Research Institute of University of Toronto is gratefully acknowledged.