Transaction Logic Programming

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

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

Abstract

An extension of predicate logic, called *Transaction Logic*, is proposed, which accounts in a clean and declarative fashion for the phenomenon of state changes in 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. The semantics leads naturally to features whose amalgamation in a single logic has proved elusive in the past. These features include both hypothetical *and* committed updates, dynamic constraints on transaction execution, non-determinism, and bulk updates. Finally, Transaction Logic holds promise as a logical model of hitherto non-logical phenomena, including so-called *procedural knowledge* in AI, and the *behavior* of object-oriented databases, especially methods with side effects. This paper presents the semantics of Transaction Logic coupled with a sound and complete proof theory for a Horn-like subset of the logic.

Appears in *Proceedings of the Tenth International Conference on Logic Programming (ICLP)*, pages 257–279. MIT Press. Conference held in Budapest, Hungary, June 21–24 1993.

*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 and a grant from New York Science and Technology Foundation. Work done during sabbatical year at the University of Toronto. Support of Computer Systems Research Institute of University of Toronto is gratefully acknowledged.