Database Programming in Transaction Logic

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

Mariano Consens
University of Toronto
Department of Computer Science
Toronto, Ontario M5S 1A4, Canada
consens@db.toronto.edu

Abstract

This paper presents database applications of the recently proposed *Transaction Logic*—an extension of classical predicate logic that accounts in a clean and declarative fashion for the phenomenon of state changes in logic programs and databases. It has a natural model theory and a sound and complete proof theory, but, unlike many other logics, it allows users to *program transactions*. In addition, the semantics leads naturally to features whose amalgamation in a single logic has proved elusive in the past. 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 focuses on the applications of $\mathcal{TR}$ to database systems, including transaction definition and execution, nested transactions, view updates, consistency maintenance, bulk updates, non-determinism, sampling, active databases, dynamic integrity-constraints, hypothetical reasoning, and imperative-style programming.

Published in *Proceedings of the fourth International Workshop on Database Programming Languages (DBPL)*, pages 309-337. Springer Verlag, Workshops in Computing Series. Workshop held in New York City, NY, 30 Aug–1 Sept, 1993.

This paper is available at the following URL:
ftp://db.toronto.edu/pub/bonner/papers/transaction.logic/dbpl93.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 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.