An Optimizing Interpreter for Concurrent Transaction Logic

for the degree of Master of Science, 2000

by

Amalia F. Sleghel

Department of Computer Science

University of Toronto

Abstract

Concurrent Transaction Logic is a new deductive database language that integrates queries, updates, and transaction composition in a simple logical framework. The language supports all the properties of classical transactions, and also the properties found in many new transaction models, such as nested transactions, concurrency within individual transaction, and fine-grained control over abort and rollback.

This thesis describes a Prolog implementation of Concurrent Transaction Logic. The prototype supports functionality in several areas including database queries, backtrackable updates, transaction definition, execution schedules, serial and concurrent execution of transactions. Two optimizations have been developed with this implementation: backtracking elimination for non-interacting processes, and query compilation. We illustrate the language and the implementation through a series of examples. The implementation is freely available on the Web.