1 A LOGIC FOR PROGRAMMING DATABASE TRANSACTIONS

Anthony J. Bonner
and Michael Kifer

Abstract: We propose an extension of classical predicate calculus, called *Transaction Logic*, which provides a logical foundation for the phenomenon of state changes in logic programs and databases. Transaction Logic comes with a natural model theory and a sound and complete proof theory. The proof theory not only verifies programs, but also executes them, which makes this logic an ideal tool for declarative programming of database transactions and state-modifying logic programs. The semantics of Transaction Logic leads naturally to features whose amalgamation in a single logic has proved elusive in the past. These features include 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 and a sound and complete SLD-style proof theory for a Horn-like subset of the logic.