Correctness of Logic Programs Constructed by a Schema-Based Method

Emmanouil Marakakis

Science Department,
Technological Educational Institute of Heraklion,
Stavromenos, Heraklion 71500, Greece
e-mail: mmarak@cs.teiher.gr

Abstract

The correctness of logic programs which are constructed by a schema-based method is presented in this paper. This schema-based method constructs typed, moded logic programs by stepwise top-down design using five program schemata, data types (DTs) and modes [1], [2]. A program is constructed by this method by successively refining undefined predicates. The lowest refinement level involves refinement by DT operations. The programs which are constructed by this method are polymorphic many-sorted programs. They are called *Schema-Instance Programs* or *SI-Programs*.

The highly structured form of SI-programs facilitates the formulation of correctness proofs. It is claimed that the structure of a logic program constructed by this schema-based method is reflected in the structure of its correctness proof. In other words, there are "proof schemes" corresponding to design schemata which are followed in the correctness proofs of logic programs. A proof scheme is proposed for each program schema.

Correctness proofs in this approach are guided by the constructed logic programs. Initially, specifications are transformed into a structured form which facilitates correctness proofs. Next, the correctness proof scheme that corresponds to the schema that has been applied for the construction of the top-level predicate is initially followed. Next, correctness proof schemes for schemata that have been applied for the construction of predicates in lower levels are followed. Correctness proofs for predicates in lower levels of an SI-program are used as lemmas for correctness proofs of predicates in higher levels of the program. Finally, correctness proofs are performed in a top-down manner, as the refinements in the program development process.

References

1. E. Marakakis, *Logic Program Development Based on Typed, Moded Schemata and Data Types*, PhD thesis, University of Bristol, February 1997.
2. E. Marakakis, J.P. Gallagher, *Schema-Based Top-Down Design of Logic Programs Using Abstract Data Types*, LNCS 883, Proceedings of Fourth International Workshops on Logic Program Synthesis and Transformation - Meta-Programming in Logic, edited by L. Fribourg and F. Turini, June 1994, Pisa, Italy, pp.138-153.