HiLog:
A Foundation for Higher-Order Logic Programming*

Weidong Chen† Michael Kifer‡ David S. Warren
Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794

Abstract

We describe a novel logic, called HiLog, and show that it provides a more suitable basis for logic programming than does traditional predicate logic. HiLog has a higher-order syntax and allows arbitrary terms to appear in places where predicates, functions and atomic formulas occur in predicate calculus. But its semantics is first-order and admits a sound and complete proof procedure. Applications of HiLog are discussed, including DCG grammars, higher-order and modular logic programming, and deductive databases.

Keywords: logic programming, higher-order logic, contextual predicate calculus, first-order semantics, resolution theorem proving, database languages, object-oriented databases, modular programming, DCG grammars.

1 Preface

Manipulating predicates, functions, and even atomic formulas is a commonplace in logic programming. For example, Prolog combines predicate calculus, higher-order and meta-level programming in one working system, allowing programmers routine use of generic predicate definitions (e.g., transitive closure, sorting) where predicates can be passed as parameters and returned as values [7]. Another well-known useful feature is the "call" meta-predicate of Prolog. Applications of higher-order constructs in the database context have been pointed out in many works, including [24, 29, 41].

Although predicate calculus provides the basis for Prolog, it does not have the wherewithal to support any of these features, which consequently have an ad hoc status in logic programming. In this paper, we investigate the fundamental principles underlying higher-order logic programming and, in particular, shed new light on why and how these Prolog features appear to work in practice. We propose a novel logic, called HiLog, which provides a clean declarative semantics to much of this higher-order logic programming.

From the outset, even the terminology of "higher-orderness" seems ill-defined. A number of works have proposed various higher-order constructs in the logic framework [1, 5, 13, 7, 23, 24, 30, 31, 42, 45] but with such a diversity of syntax and semantics, it is not always clear what kind of higher-orderness is being claimed. In our opinion, there are at least two different facets to the issue: a higher-order

*This paper is an expanded version of the work previously reported in [10, 11].

†Present address: Computer Science and Engineering, Southern Methodist University, Dallas, TX 75275.

‡Work of M. Kifer was partially supported by the NSF grant IRI-8903507.