DATABASE LOGIC PROGRAMMING

MOSHE Y. VARDI *

The late 1970s saw the emergence of two new research areas in computer science: *deductive databases* and *logic programming*, both rooted in earlier research on theorem proving. Research in deductive databases is concerned with extending database systems with deductive capabilities, and research in logic programming is concerned with the use of logic as a programming language. The relationship of these areas to each other was recognized at a workshop held in Toulouse, France, in 1977, which culminated in the book *Logic and Databases* (edited by H. Gallaire and J. Minker).[1] Further developments of these intimately related areas in the 1980s were intertwined. The new area of *logic database programming*, which can be viewed as the meeting point of logic programming and deductive databases, is concerned with the use of logic as a programming language to extend database system capabilities.

The five papers in this special issue address the three major aspects of database logic programming: expressiveness, optimization, and complexity. The first paper, "Set Constructors in a Logic Database Language", by C. Beeri, S. Naqvi, O. Shmueli, and S. Tsur, describes a logic-programming language with set constructors. The paper investigates how semantics can be defined for the language, and, more broadly, it examines the intricacies arising from the introduction of set constructors into a logic-based language.

The next two papers deal with optimization of database logic programs. The paper "A Simple Characterization of Uniform Boundedness for a Class of Recursions", by J. F. Naughton and Y. Sagiv, deals with the detection of bounded recursion (i.e., recursion that has a data-independent bound on its depth). This is a powerful optimization technique, as bounded recursion can be replaced by equivalent nonrecursive rules. The paper identifies a class of programs for which boundedness can be detected in polynomial time. The paper "On the Power of

*Special Issue Editor.

Address correspondence to Moshe Y. Vardi, IBM Almaden Research Center K53/802, 650 Harry Road, San Jose, CA 951020-6099.

[1] For a historical perspective on this development, see "Perspectives in Deductive Databases" by J. Minker, in *J. Logic Programming* 5:33–60 (1988).

655 Avenue of the Americas, New York, NY 10010 0743-1066/91/$3.50

Magic", by C. Beeri and R. Ramakrishnan, describes a general optimization technique, the *magic set* method. In this method the original program is rewritten to take advantage of information passing among subgoals.

The last two papers deal with the complexity of evaluating database logic programs. The paper "Efficient Evaluation for a Subset of Recursive Queries", by G. Grahne, S. Sippu, and E. Soisalon-Soininen, considers the classes of regular programs and of linear programs. It develops an efficient algorithm for evaluating such programs, using well-known results on graph traversals. The paper "Comparison of Methods for Logic Query Implementation", by A. Marchetti-Spaccamela, A. Pelaggi, and D. Saccà, considers the class of canonical strongly linear programs. It provides a worst-case complexity analysis for three well-known methods for evaluating such programs.

The authors of these papers were invited to submit their papers to this special issue. All the papers were subjected to the normal referring procedure of this journal. I thank the authors and the anonymous referees for their diligence in making this special issue possible.