Stan Raatz

Graph-Based Proof Procedures for Horn Clauses

1990 Springer Science+Business Media, LLC

Stan Raatz
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903
USA

Printed on acid-free paper.

ISBN 978-0-8176-3530-5 ISBN 978-1-4899-3556-4 (eBook)
DOI 10.1007/978-1-4899-3556-4

Camera-ready copy prepared by the author.

9 8 7 6 5 4 3 2 1

PREFACE

The origins of this monograph lie in my Ph.D. dissertation of 1987 at the University of Pennsylvania, which was concerned with proof procedures for the Horn clause subset of logic. The rise of logic programming has made this an important area of study. All Prologs are based on a variant of resolution, and inherit various properties related to this proof method. This monograph studies the paradigm of logic programming in the context of graph-based proof procedures which are unrelated to resolution.

The monograph is not a general introduction to logic programming, although it is self-contained with respect to the mathematics used. It should appeal to the computer scientist or mathematician interested in the general area we now call computational logic. A large part of the monograph is devoted to detailed proofs that the methods we present are sound and complete, which in the context of the logic programming, means that the operational and denotational semantics agree.

The monograph is organized as follows. After a chapter on mathematical preliminaries, in chapter 3 we present examples of logic programs and define a relevant semantics. The usual semantics of Prolog is not admited by the methods studied in this work. In chapter 4 we present the mechanics of the first method, Hornlog, which admits one-sorted Horn clause programs consisting of any arbitrary Horn clauses, including clauses of the form $\leftarrow B_1, \ldots, B_n$, and queries of the form $Q = \exists x_1 \ldots \exists x_n(\neg H_1 \vee \cdots \vee \neg H_m)$ where $\{H_1, \ldots, H_m\}$ are again one-sorted Horn clauses. In chapter 5, we give constructive proofs of the soundness and completeness of the answer substitution, and show the relationship between the operational semantics of the method and the model-theoretic semantics of the underlying language.

In chapters 6 and 7, we define an equational extension of Hornlog, the $HE^{\dagger}$-refutation method, which applies to many-sorted first-order equational Horn clause programs consisting of clauses of the form $s \doteq t$, $A \leftarrow B_1, \ldots, B_n$, or $\leftarrow B_1, \ldots, B_n$ where s and t are first-order terms, A is a non-equational atomic formula, and $B_1, \ldots, B_n$ are either equational or non-equational atomic formulae. This class of programs subsumes the paradigms of functional, logic, and equational programming. In chapter 8, we show the soundness and completeness of this method. Finally, in the appendix chapter 9 we consider the design of an implementation on a abstract parallel machine.

Acknowledgements

I would first like to thank my Ph.D. advisor, Jean Gallier, for the encouragement, personal support, and technical advice he offered at critical moments in the course of this work. He was, in addition, a good friend and who helped me through the inevitable ups and downs encountered along the way. I would also like to thank the members of my Ph.D. committee, Peter Buneman, Tim Finin, Jean-Yves Girard, Jean-Louis Lassez, Ken McAloon, and Dale Miller for taking the time to evaluate this work. Their suggestions improved it considerably.

Finally, let me express my thanks and appreciation to my parents for their unquestioning support of this endeavor, and to my friends in philosophy, Charan, Mary, and Doug, for their help in the search for life's meaning.

Table of Contents