ATLANTIS:

A TOOL FOR LANGUAGE DEFINITION

AND INTERPRETER SYNTHESIS

Michael John Oudshoorn, B.Sc.(Hons.)

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF ADELAIDE

August 1992

Abstract

Programming language semantics are usually defined informally in some form of technical natural language, or in a very mathematical manner with techniques such as the Vienna Definition Method (VDM) or denotational semantics. One difficulty which arises from serious attempts to define language semantics is that the resulting definition is generally suitable for a single limited kind of reader. For example, the more formal kind of definition may suit a compiler writer or a language designer, but will be less convenient for other potential classes of reader, such as programmers. The latter frequently make use of some completely separate description (e.g., an introductory text book on the language); not surprisingly, inconsistencies between these separate descriptions and the language definition are commonplace.

This thesis develops a technique for the definition of programming language semantics which is suitable for a wide range of potential readers. This technique employs an operational semantic model which is based on the algebraic specification of abstract data types; the semantic model manipulates multi-layer descriptions of language semantics and supports multiple passes in these descriptions.

The semantic technique described in this thesis lends itself to the semi-automatic generation of an interpreter from the language definition, a fact which acts as an

incentive to language designers to produce a formal definition of any new programming language, since the prototype implementation allows experimentation with new language features and their semantics. The system which generates an interpretive implementation from a language definition is called ATLANTIS, A Tool for LANguage definiTion and Interpreter Synthesis, and is also described in this thesis.

Declaration

This is to certify that this thesis contains no material which has previously been accepted for the award of any degree or diploma in any University. To the best of my knowledge and belief it contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

If this thesis is accepted for the award of the degree, permission is granted for it to be made available for loan and photocopying.

Michael Oudshoorn

August 1992

Acknowledgments

I wish to acknowledge all the help and encouragement that I have received throughout my candidature as a Ph.D. student. In particular, I am indebted to my supervisor, Prof. Chris Marlin, for his advice, thoughts, guidance and encouragement; and to Dr. David Bover who acted as my supervisor whilst Prof. Marlin was on study leave during the second half of 1988 and early 1989. I also wish to express my thanks to Prof. Peter King and Dr. Brian Molinari for their comments on this thesis. I am also indebted to my wife, Jo, and my son, Daniel, for having to tolerate many late nights, and often having to come second behind my study. Many thanks to my parents, Bep and Nick, who supported and encouraged my study.

I also wish to thank the other members of the staff, the postgraduate students and visiting speakers, who have all contributed to this thesis in many small ways. Of particular note in this regard are Michael McCarthy and Ali Mili. I also wish to acknowledge the contribution of Keith Ransom, who developed the Ada code in Section 5.5 and carried out the experiments referred to in Section 5.7.

I would also like to thank all those companies that allowed me time on their machines and gave me access to their Pascal compilers to obtain the results discussed in Chapter 1; I have been asked by many of these companies not to publish their names.

Contents

List of Tables

List of Figures