ARTIFICIAL INTELLIGENCE SERIES

7

Logic for Problem Solving

Robert Kowalski
Imperial College of Science and Technology
University of London

NORTH-HOLLAND
NEW YORK • AMSTERDAM • OXFORD

Elsevier Science Publishing Co., Inc.
52 Vanderbilt Avenue, New York, New York 10017

Sole Distributors Outside USA and Canada:

Elsevier Science Publishers B.V.
P.O. Box 211, 1000 AE Amsterdam, The Netherlands

Library of Congress Cataloging in Publication Data

Kowalski, Robert.
Logic for problem solving.
(Artificial intelligence series) (The Computer science library)

Bibliography: p.
Includes index.
1. Problem solving. 2. Electronic digital computers—Programming.
3. Logic, Symbolic and mathematical. I. Title.
QA63.K68 519.7 79-22659
ISBN 0-444-00365-7 (hbk.)
ISBN 0-444-00368-1 (pbk.)

Current printing (last digit)
10 9 8

Manufactured in the United States of America

To

my parents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Preface

This book investigates the application of logic to problem-solving and computer programming. It assumes no previous knowledge of these fields, and may be appropriate therefore as an introduction to

logic,
the theory of problem-solving, and
computer programming.

Logic

Logic is an important tool in the analysis and presentation of arguments. It investigates whether assumptions imply conclusions, independently of their truth or falsity and independently of their subject matter. This book aims to apply the traditional methods of logic to contemporary theories of problem-solving and computer programming.

As an introduction to logic, the book differs from others in its use of the clausal form of logic. This has several advantages. Clausal form is simpler than the standard form of logic but is just as powerful. It is simple enough to be introduced directly, without the usual preliminary study of propositional logic, and it bears greater resemblance than standard form to other formalisms used in data processing and computer programming.

This book is not concerned with the mathematics of logic but with its applications. For an interesting and more thorough discussion of the relationships between logic and language the reader is advised to consult the books by Quine [1941] and Hodges [1977].

Problem-solving

The clausal form of logic can be used to elucidate and compare models of problem-solving developed in cognitive psychology and artificial intelligence. This book investigates the heuristic search, problem-reduction and program execution models of problem-solving and argues that logical inference provides a model which is both simpler and more powerful.

The interpretation of logical inference as problem-solving builds upon the distinction between bottom-up reasoning, forward from assumptions to conclusions, and top-down reasoning, backwards from goals to subgoals.

Problem-solving

The problem-solving interpretation of inference is primarily the top-down interpretation. Bottom-up inference is the manner in which solutions are generally presented and justified, whereas top-down inference is the manner in which solutions are most often discovered. Bottom-up inference is the synthesis of new information from old; top-down inference is the analysis of goals into subgoals.

This book covers similar ground to the problem-solving sections of the books by Nilsson [1971], Winston [1977] and Bundy et al [1978]. Where those books use production systems, LISP or LOGO as the unifying formalism, ours uses the clausal form of logic.

Computer programming

Employed as a language for communicating with computers, logic is higher-level and more human-oriented than other formalisms specifically developed for computers. In contrast with conventional computing methodology, which employs different formalisms for expressing programs, specifications, databases, queries and integrity constraints, logic provides a single uniform language for all of these tasks. We shall investigate the use of logic for databases, but concentrate on its use as a programming language.

The meaning of programs expressed in conventional languages is defined in terms of the behaviour they invoke within the computer. The meaning of programs expressed in logic, on the other hand, can be defined in machine-independent, human-oriented terms. As a consequence, logic programs are easier to construct, easier to understand, easier to improve, and easier to adapt to other purposes.

The same methods of top-down inference which give logic a problem-solving interpretation can be used to execute logic programs efficiently by means of computers. Top-down inference unifies problem-solving and computer programming. Moreover, it provides many of the facilities for intelligent program execution, such as non-determinism, parallelism, and procedure call by pattern-matching, which are under development for more conventional programming languages today. An efficient programming language, called PROLOG [Colmerauer et al 1972], [Roussel 1975], [Bruynooghe 1976], [Warren, Pereira and Pereira 1977] and [Clark and McCabe 1979], based on the clausal form of logic, has been used for applications in artificial intelligence, databases and engineering.

Mechanical theorem-proving

The use of the clausal form of logic and its associated systems of inference is based upon investigations into the mechanical proof of theorems by means of computers. The resolution rule of Robinson [1965a] and the model-elimination proof procedure of Loveland [1968, 1969] have been the main antecedents of the inference systems investigated in this book. Their inference methods in turn are based upon earlier researches by Herbrand [1930] and Prawitz [1960].

direction of change. This combines the problem-solving interpretation of logic with the classical use of logic in the analysis of human knowledge and belief.

Level of the book

This book is an extension of lecture notes prepared in March 1974 [Kowalski 1974b] for an advanced course on the Foundations of Computer Science held at the Mathematics Centre in Amsterdam. Short courses on the same material were given by the author in Edinburgh, Milan, Rome and Stockholm, between 1973 and 1975. Since 1975, parts of the book have been used for introductory courses in logic and in problem-solving given to computing students at Imperial College. A complete course covering all the material in the book was given at the University of Syracuse in 1978.

The book is written at an informal level and contains almost no proofs. It assumes no previous background in logic, problem-solving or computer science, and may be suitable,therefore, for students at the first year undergraduate level. Many of the exercises, however, are of a more advanced level. Moreover, some of the discussion in Chapter 5, comparing logic with conventional programming languages, may not be completely intelliigible to readers without previous programming experience.

Acknowledgements

Much of the material in this book has been influenced by the work of my colleagues Keith Clark, Alain Colmerauer, Pat Hayes, Maarten van Emden and David Warren. I am grateful to them and to Frank Brown, Alan Bundy, Tony Hoare, Wilfred Hodges, Chris Hogger, Jan Nilsson, George Pollard, Ray Reiter, Richard Waldinger and George Winterstein, for the valuable comments they have made on earlier drafts of the book, and to Karen King, Frank McCabe, Kevin Mitchell and Chris Moss, for helping to produced the camera-ready copy. I am also happy to acknowledge the support of the Science Reseach Council.

I am especially indebted to my wife, Danusia, and children, Dania, Tania and Janina, for their patience and encouragement.

Preface

Although the inference methods in this book were originally designed for use by computers, they can also be used by human beings. The problem-solving strategies developed for efficient mechanical theorem-proving are similar to those investigated by researchers concerned with computer simulation of human problem-solving. In particular we have attempted to present a view of logic which reconciles the machine-oriented view of resolution with the heuristic proof-procedures of Bledsoe [1971, 1977] and his colleagues.

This book can be regarded as a text in the field of mechanical theorem-proving, similar to those by Chang and Lee [1973], Loveland [1978] and Robinson [1979]. It is less formal, however, and makes no attempt to give a broad coverage of the field.

Organisation of the book

The book is organised into three parts. The first part, Chapters 1 and 2, deals with the machine-independent semantics of the clausal form of logic and the use of clausal form for representing information; the second part, Chapters 3 to 8, deals with inference systems for clausal form; and the third , Chapters 9 to 13, investigates extensions of clausal form as well as more powerful problem-solving methods.

The first part of the book emphasises that logic, unlike most other formalisms, can be understood without understanding its behaviour. Examples are given of the use of logic for describing programs and databases, and clausal form is compared with semantic networks for representing the meanings of natural language sentences.

The second part of the book introduces inference methods for clausal form in stages of increasing complexity. Chapters 3 to 6 deal with inference methods for Horn clauses, which are simplified sentences, mainly of the form

$$A \text{ if } B_1 \text{ and } B_2 \text{ and } \ldots \text{ and } B_m.$$

Top-down and bottom-up inference are introduced in Chapter 3 as generalisations of top-down and bottom-up parsing procedures for context-free grammars. Chapter 4 deals with the problem-solving interpretation of top-down inference, whereas Chapter 5 deals with its programming language interpretation. Chapter 6 describes the application of Horn clause logic to plan-formation problems. Inference methods for non-Horn clause problems and their problem-solving interpretation are investigated in Chapters 7 and 8.

Chapter 9 deals with global problem-solving methods for clausal form, whereas the remaining chapters investigate various extensions of clausal form. Although clausal form is as powerful as the standard form of logic, it is sometimes less natural. The standard form of logic and its relationship to clausal form are investigated in Chapter 10. Definitions using "if-and-only-if" are treated separately in Chapter 11. In Chapter 12 we consider an extension of logic which combines the use and mention of sentences in a manner similar to that of natural language. The final chapter deals with the dynamics of changing information systems, paying special attention to the role of contradiction in determining the