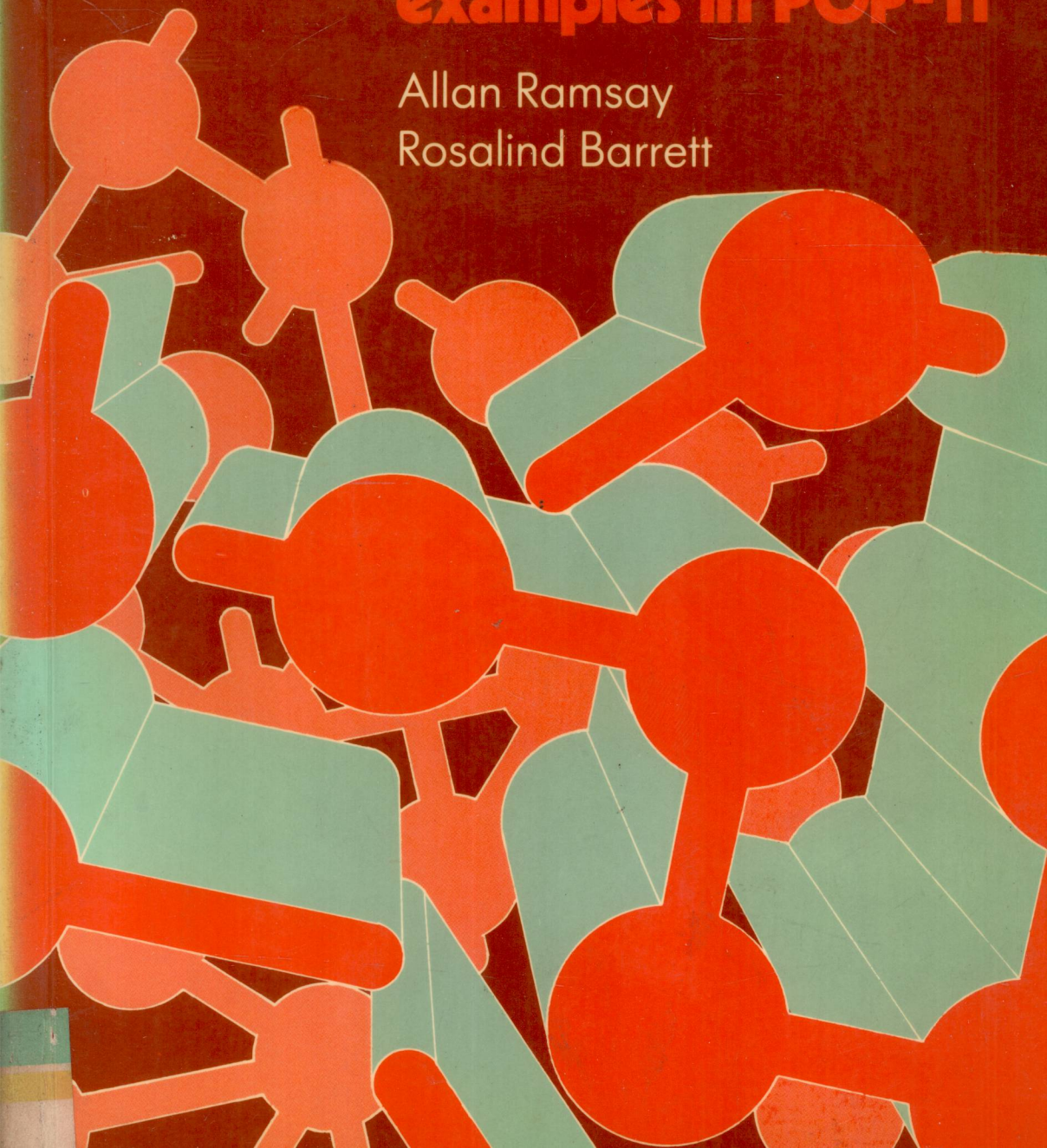
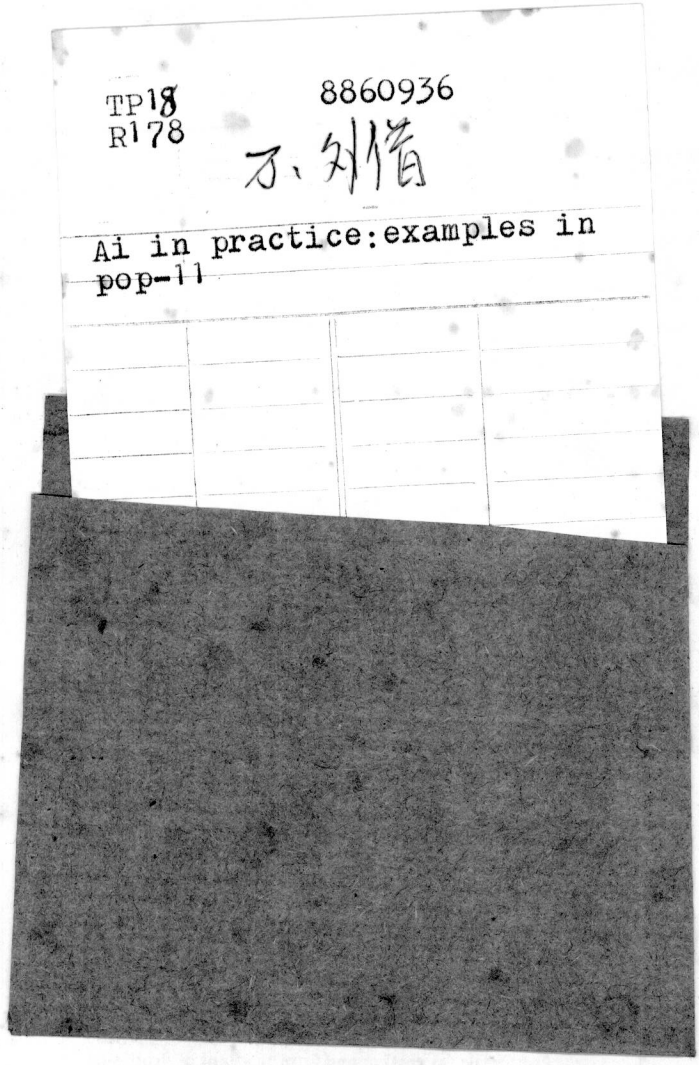
Ellis Horwood Series in ARTIFICIAL INTELLIGENCE

AI IN PRACTICE

examples in POP-11

Allan Ramsay
Rosalind Barrett





AI IN PRACTICE: Examples in POP-11

ELLIS HORWOOD BOOKS IN COMPUTING SCIENCE

General Editors: Professor JOHN CAMPBELL, University College London, and BRIAN L. MEEK, King's College London (KQC), University of London

Series in Artificial Intelligence

Series Editor: Professor JOHN CAMPBELL, Department of Computer Science, University College London

Abramsky, S. & Hankin, C. (editors) — **Abstract Interpretation of Declarative Languages***
Boguraev, B.K. — **Natural Language Interfaces to Computational Systems***
Bramer, M.A. (editor) — **Computer Game Playing: Theory and Practice**
Briscoe, E.J. — **Computational Speech Processing: Syntax and Prosody***
Campbell, J.A. (editor) — **Implementations of PROLOG**
Campbell, J.A., Forsyth, R., Narayanan, A. & Teague, M. — **Dictionary of Artificial Intelligence***
Davies, R. (editor) — **Intelligent Information Systems: Progress and Prospects**
Elcock, E.W. & Michie, D. (editors) — **Machine Intelligence 8: Machine Representations of Knowledge**
Evans, J.B. — **Discrete System Simulation***
Forsyth, R. & Rada, R. — **Machine Learning: Applications in Expert Systems and Information Retrieval**
Gabbay, D.M. — **Programming in Pure Logic***
Gergely T. & Futó, I. — **Artificial Intelligence in Simulation***
Glicksman, J. — **Image Understanding and Machine Vision***
Gottinger, H.W. — **Artificial Intelligence: The Commercialisation of Intelligent Systems***
Hawley, R. (editor) — **Artificial Intelligence Programming Environments***
Hayes, J.E., Michie, D. & Mikulich, L.I. (editors) — **Machine Intelligence 9: Machine Expertise and the Human Interface**
Hayes, J.E., Michie, D. & Pao, Y.-H. (editors) — **Machine Intelligence 10: Intelligent Systems: Practice and Perspective**
Hayes, J.E. & Michie, D. (editors) — **Intelligent Systems: The Unprecedented Opportunity**
Hunter, J.R.W., Gotts, N.M. & Sinnhuber, R.K.E.W. — **Artificial Intelligence in Medicine***
Łukaszewicz, W. — **Nonmonotonic Reasoning***
Mellish, C. — **Computer Interpretation of Natural Language Descriptions**
Michie, D. — **On Machine Intelligence, Second Edition**
Partridge, D. — **Artificial Intelligence: Applications in the Future of Software Engineering**
Ramsay, A. & Barrett, R. — **AI in Practice: Examples in POP-11**
Slatter, P.E. — **Cognitive Emulation in Expert Systems***
Savory, S.E. — **Artificial Intelligence and Expert Systems***
Spacek, L. — **Advanced Programming in PROLOG***
Sparck Jones, K. & Wilks, Y. (editors) — **Automatic Natural Language Parsing**
Steels, L. & Campbell, J.A. (editors) — **Progress in Artificial Intelligence**
Torrance, S. (editor) — **The Mind and the Machine**
Turner, R. — **Logics for Artificial Intelligence**
Wallace, M. — **Communicating with Databases in Natural Language**
Waterworth, J.A. — **Speech and Language-based Communication with Machines***
Wertz, H. — **Automatic Correction and Improvement of Programs**
Yazdani, M. (editor) — **New Horizons in Educational Computing**
Yazdani, M. & Narayanan, A. (editors) — **Artificial Intelligence: Human Effects**

** In preparation*

AI IN PRACTICE: Examples in POP-11

ALLAN RAMSAY, B.Sc., M.Sc., Ph.D.
Lecturer in Artificial Intelligence
Cognitive Studies Programme, University of Sussex

and

ROSALIND BARRETT, B.A.
Technical Author
Cognitive Studies Programme, University of Sussex

ELLIS HORWOOD LIMITED
Publishers · Chichester

Halsted Press: a division of
JOHN WILEY & SONS
New York · Chichester · Brisbane · Toronto

First published in 1987 by
ELLIS HORWOOD LIMITED
Market Cross House, Cooper Street,
Chichester, West Sussex, PO19 1EB, England
The publisher's colophon is reproduced from James Gillison's drawing of the ancient Market Cross, Chichester.

Distributors:
Australia and New Zealand:
JACARANDA WILEY LIMITED
GPO Box 859, Brisbane, Queensland 4001, Australia
Canada:
JOHN WILEY & SONS CANADA LIMITED
22 Worcester Road, Rexdale, Ontario, Canada
Europe and Africa:
JOHN WILEY & SONS LIMITED
Baffins Lane, Chichester, West Sussex, England
North and South America and the rest of the world:
Halsted Press: a division of
JOHN WILEY & SONS
605 Third Avenue, New York, NY 10158, USA

British Library Cataloguing in Publication Data
Ramsay, Allan, *1953*–AI in practice: examples in POP–11. —
(Ellis Horwood books in computing science: series in artificial intelligence)
1. Artificial intelligence — Data processing
2. POP–11 (Computer program language)
I. Title II. Barrett, Rosalind
006.3'028'55133 Q336

Library of Congress CIP Data also available

ISBN 0–7458–0063–7 (Ellis Horwood Limited — Library Edn.)
ISBN 0–7458–0167–6 (Ellis Horwood Limited — Student Edn.)
ISBN 0–470–20770–1 (Halsted Press)

Printed in Great Britain by R.J. Acford, Chichester

CONTENTS

PREFACE

Introduction

This book is intended for people who have a theoretical understanding of Artificial Intelligence (AI), but who have little experience of the problems that arise in the practical application of this understanding. The book contains comprehensive illustration of programs demonstrating advanced AI techniques. By complementing the theory with clearly commented programs, we hope to help our readers gain a much deeper understanding of some of the major techniques in current use in AI.

The programs in this book demonstrate state-of-the-art techniques. We have tried to steer a course between toy programs, from which the reader would learn little, and experimental programs which have not been thoroughly tested. The more advanced concepts which we make use of include such things as non-linear, hierarchical planning, theorem proving for modal logic, rule compilation for expert systems, line finding based on local computation, and chart parsing for functional unification grammar. These are all ideas which are well known in the AI literature, but for which concrete implementations are seldom presented.

The programs are written in the high-level programming language POP-11. POP-11 is the core language of the POPLOG

system - an integrated, interactive software development environment containing incremental compilers for POP-11, LISP and PROLOG, all written entirely in POP-11. It was, however, originally developed as a language for AI applications. We hope that this book will show how well suited it is to its original function, as well to the sort of systems work that has been done in it for POPLOG.

The book, *POP-11: A Practical Language for Artificial Intelligence*, (Barrett, Ramsay and Sloman, 1985) is the text teaching the language. Readers with access to POP-11 who are familiar with all the facilities presented there are encouraged first to use the programs developed in this book, and later to modify them and develop their own programs in order to explore more areas in AI. Any POP-11 procedures not covered in the teaching text are described in an appendix at the end of this book.

Readers unfamiliar with POP-11 itself, but literate in other high-level programming languages, such as LISP, PROLOG, C and PASCAL, will easily be able to follow the examples but will not be able to develop them.

The structure of the book

The book is divided into five self-contained chapters representing major areas in AI. They are Planning, Theorem Proving, Expert System Shells, Vision and Natural Language. Their order in the book is of no significance, and nothing in any one chapter depends on anything in another. Readers need only use chapters which are of interest to them. The appendices at the back of the book reference all the POP-11 procedures used, describe any POP-11 facilities not covered by Barrett, Ramsay and Sloman (1985), and provide useful addresses for finding out more about POP-11.

Accessing POP-11

POP-11 is currently only available as part of POPLOG. It is designed to be portable and runs on VAX computers under VMS, Berkeley UNIX and Ultrix operating systems, GEC Series 63 computers under AT&T UNIX System V, M68000 computers under Unisoft UNIX, SUN.2 and SUN.3 under Berkeley UNIX, and Hewlett Packard M68000 workstations also running Berkeley UNIX. POP-11 can be used with an ordinary VDU with screen editing capabilities, and can also be linked to a window/mouse mechanism on the SUN workstation and other workstations with such devices. POPLOG is marketed

commercially by Systems Designers.

Towards the end of 1986 Cognitive Applications will release POP-11 on the Apple Macintosh. During 1987 it will be ported onto the IBM-PC and other popular micro-computers.

Acknowledgements

The programs in this book are based on programs developed by colleagues in the Cognitive Studies Program and the Experimental Psychology Laboratory at the University of Sussex. They have undergone many modifications since we selected them from the many programs that were available to us, and any bugs they contain now are our responsibility rather than their original authors'. Many people have contributed to the development of these programs over the past few years, and in some cases the history of authorship is not entirely clear. We do know that the following people made substantial contributions, and we would like to thank them for allowing us to use their work: Steve Hardy, Aaron Sloman and David Hogg for parts of the chapter on vision; Steven Isard, Roger Evans and Chris Mellish for parts of the chapter on language processing; Chris Mellish for most of the code for the expert system shells; Steve Hardy for the basis of the planner; and John Gibson for much of the description of the virtual machine.

1

PLANNING

1.1 INTRODUCTION

This chapter develops a non-linear, hierarchical planner. For illustration we use the planner in the 'blocks world', the traditional toy domain for AI programs, but the planner is not restricted to this world. It will work for any domain where actions can be described in terms of discrete lists of preconditions (facts which must be true for the action to be performed) and effects (facts which will be true after the action has been performed).

Our planner describes actions in terms of preconditions and effects. It is given a list of facts which are currently true and a list of goals which we would like to be true, and attempts to produce a sequence of actions which will bring about the list of goals. It uses a combination of backward chaining, as in the original STRIPS planner (Fikes and Nilsson 1971) and heuristic search. The backward chaining is used for providing plans for each of the individual top level goals. The heuristic search, which is reminiscent of the strategy used by NOAH (Sacerdoti 1977), is used for working out what order to tackle the main goals in. Before we go on to consider the mechanisms behind the planner in detail we will look at an example of the

sort of plan it generates.

The blocks world consists of a number of blocks, a table, and a robot hand. The blocks are either on the table, on each other, or held by the hand somewhere above the table or a pile of blocks. At most one block can be on any other, though any number may form a stack, and at most one block can be held in the hand at any one time. The table is divided into places filled by a block, and spaces. The goal of the system is to find a plan to arrange the blocks into positions which you specify. You can grasp and ungrasp blocks, and move, raise or lower either an empty hand or a hand holding a block. You can only grasp blocks which have clear tops. The state of the world is represented by a list of facts. We use the following state of affairs for illustrative purposes:

```
[[object table] [object b1] [object b2]
 [object b3] [object b4]
 [block b1] [block b2] [block b3] [block b4]
 [on b1 table] [position b1 [1 1]]
 [on b2 b1] [position b2 [1 1]]
 [holding b3] [position b3 [1 1]]
 [position hand [1 1]]
 [position space [20 20]]
 [position space [30 30]]
 [on b4 table] [position b4 [10 10]]]
```

This can be represented pictorially as in Fig. 1.1. If we give the planner

```
[[on b1 b3] [on b3 b2]]
```

as a list of goals, we can represent the desired state of the world pictorially as in Fig. 1.2.

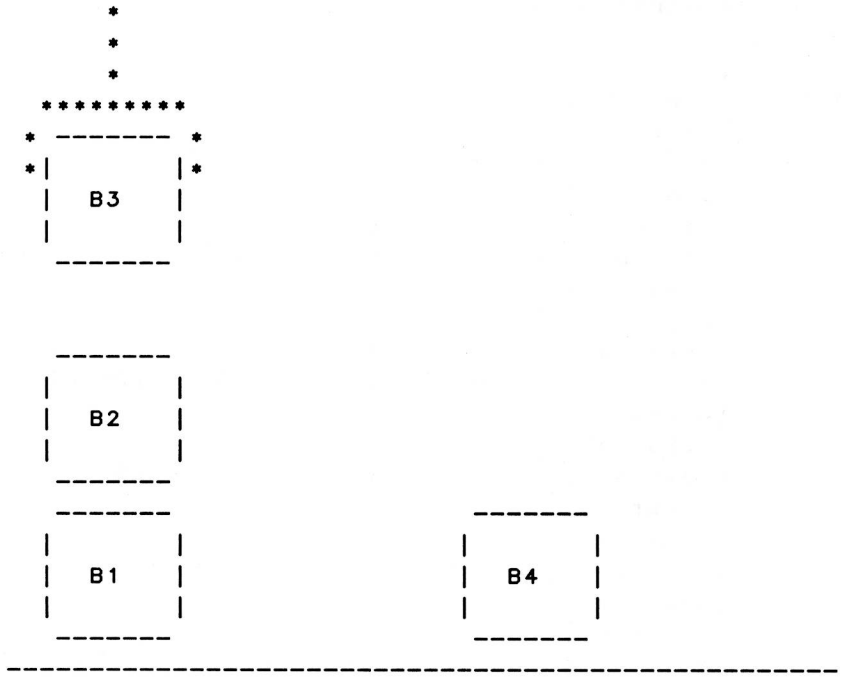


Figure 1.1 Initial blocks world

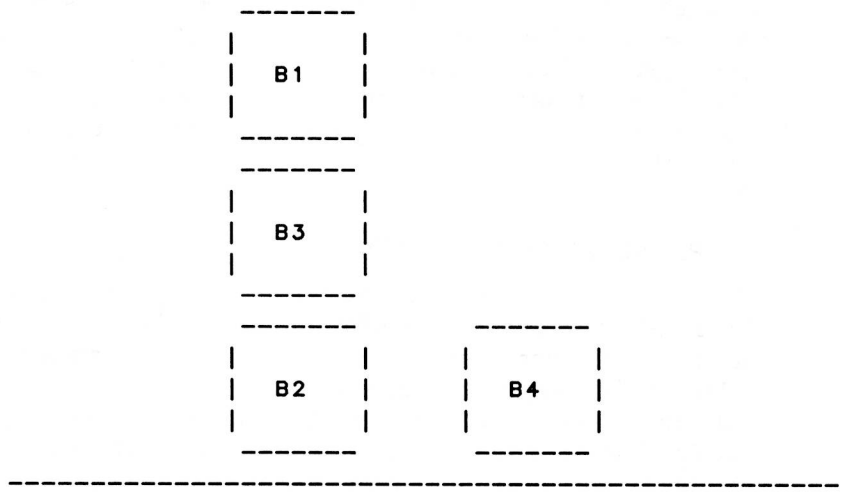


Figure 1.2 Final blocks world

The planner, called `make_plan`, would produce the following:

```
[move hand ( holding b3 ) from [1 1] to [20 20]]
[lower b3]
[ungrasp b3]
[move hand ( empty ) from [20 20] to [1 1]]
[grasp b2]
[raise b2]
[move hand ( holding b2 ) from [1 1] to [30 30]]
[lower b2]
[ungrasp b2]
[move hand ( empty ) from [30 30] to [1 1]]
[move hand ( empty ) from [1 1] to [20 20]]
[grasp b3]
[raise b3]
[move hand ( holding b3 ) from [20 20] to [30 30]]
[lower b3]
[ungrasp b3]
[move hand ( empty ) from [30 30] to [1 1]]
[grasp b1]
[raise b1]
[move hand ( holding b1 ) from [1 1] to [30 30]]
[lower b1]
```

This is a fairly complex plan by the standards of AI planning systems. It contains one minor hiccough, about half-way through, where it does two consecutive `moves`. This happened because of the way that the plans for achieving the two main goals get spliced together. Apart from this the plan is as efficient as it could be, and was discovered in an acceptable amount of time (just over 70 CPU seconds on a DEC VAX-11/750).

1.2 REPRESENTING ACTIONS

We represent the operators, or the description of the actions, as 'schemas'. The result of performing an action is to change the state of the world as it is recorded in the database. This is achieved by simply adding and deleting facts to and from the database. We make a small change to the traditional presentation of actions. Rather than having add- and delete-lists of facts we have a single effects-list which may contain negated facts. This makes little difference to the description of the effects of an action, but it does enable us to be more