
AI and Expert Systems

A Comprehensive Guide to Turbo Pascal



AI and Expert Systems

A Comprehensive Guide to Turbo Pascal

Robert I. Levine

Senior Research Engineer, Unisys Corporation

Diane E. Drang

*Director of Educational Services, Jewish Board
Family & Children's Services*

Barry Edelson

Editor and author

Second Edition

McGraw-Hill Publishing Company

New York St. Louis San Francisco Auckland Bogotá
Caracas Hamburg Lisbon London Madrid Mexico
Milan Montreal New Delhi Oklahoma City
Paris San Juan São Paulo Singapore
Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Levine, Robert I.

AI and expert systems: A comprehensive guide to Turbo Pascal.

Bibliography: p.

Includes index.

I. Artificial intelligence. 2. Expert systems
(Computer science) 3. Microcomputers. 4. Turbo
Pascal (Computer program) I. Drang, Diane E.

II. Edeleon, Barry. III. Title.

Q335.L473 1990 006.3

ISBN 0-07-037495-3

Copyright © 1990, 1988 by McGraw-Hill, Inc. All rights reserved.
Printed in the United States of America. Except as permitted
under the United States Copyright Act of 1976, no part of this
publication may be reproduced or distributed in any form or by
any means, or stored in a data base or retrieval system, without
the prior written permission of the publisher.

234567890 DOC/DOC 9876543210

ISBN 0-07-037495-3

The editors for this book were Theron Shreve and Nancy Young,
the designer was Naomi Auerbach, and the production supervisor
was Dianne L. Walber. It was set in Century Schoolbook
by Professional Composition, Inc.

Printed and bound by R. R. Donnelley & Sons Company.

Information contained in this work has been obtained by McGraw-Hill, Inc., from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantees the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

*For more information about other McGraw-Hill materials,
call 1-800-2-MCGRAW in the United States. In other
countries, call your nearest McGraw-Hill office.*

Preface

Welcome to the inner workings of the brain. Let us take an odyssey into the understanding of how we learn, think, and reason. We will take you into the world of expert systems and neural networks, the second generation of artificial intelligence. Over the next few years, neural network programming techniques will invade many areas presently limited by expert system knowledge. The ability to increase domain knowledge beyond that of experts is necessary for the rapid advancement of artificial intelligence. Neural networks are already showing up in such areas as speech and vision systems. As artificial intelligence which contains expert systems and neural networks reaches more and more areas of daily living, the need to understand and the ability to apply the concepts will become more of a necessity.

This book introduces and explains the concepts of neural networks and advanced artificial intelligence in a language and a set of pictures that everyone can understand. We will take you into a more advanced world of artificial intelligence, showing you how to use many methods (multiple paradigms) for solving problems. Then we will bring all the concepts together by giving you an understanding of object oriented systems. These concepts will enable you to approach the design of both simple and advanced systems.

This book teaches applied concepts. All the programming you will see in these pages can be implemented on a first-generation personal and home computer. Short programs, written in Pascal, are provided in each of the chapters that contain concepts. These programs are easy to use and understand.

This book provides you with a simple uncomplicated view of the first and second generation of artificial intelligence. It has been carefully planned so that each chapter is self-contained. This technique allows you to learn concepts without having to make frequent references to previous chapters. The combination of this technique, a simple approach, and examples that will appeal to a wide range of people makes this book required reading for anyone trying to understand the basic

concepts. It is designed to be used by advanced students, corporate managers, and programmers as well as noncomputer people. There are many step-by-step illustrations and more than 50 examples. These are designed to make the subject matter very easy to understand.

Chapter 1 provides the reader with an overview of the concept of intelligence and how this concept can be identified.

Chapter 2 shows how this concept can be transferred to a computer to create artificial intelligence.

Chapter 3 discusses the concept of expert systems and its relationship to artificial intelligence.

Chapter 4 provides a brief and simple overview of natural language processing.

Chapter 5 discusses forward chaining, providing an extensive set of examples and a forward chaining tool written in Pascal. The tool can be used in a limited way to experiment with the concepts; it can also be easily modified into a more powerful structure.

Chapter 6 discusses backward chaining in a way that is similar to Chapter 5. A backward chaining tool is also provided in this chapter.

Chapter 7 discusses bayesian probability and fuzzy conditions as they apply to expert systems. A programming example illustrating how the concepts can be applied to a rule-based system is provided.

Chapter 8 provides a discussion of the design of a truncated expert system for personal finance. This topic applies to anyone who has money dealings.

Chapter 9 provides a discussion of the design of a truncated expert system for optimizing a sales effort with a customer. This topic applies to all corporations and people since everyone is always selling something.

Chapter 10 discusses the design of an expert system that diagnoses learning problems. This is of interest to schools, corporations analyzing employee capabilities, concerned parents, and individuals.

Chapter 11 discusses object-oriented programming as it applies to expert systems. A set of examples which clearly illustrate some of the concepts is provided. An object-oriented system is written in Pascal to allow you to experiment with the concepts.

Chapter 12 discusses an engineering application using object-oriented programming. A worked-out example is provided.

Chapter 13 provides an expert system example that issues flood warnings using object-oriented programming.

Chapters 14, 15, and 16 discuss semantic nets, certainty factors, and automated learning. A program in Pascal is provided for each of these topics.

Chapter 17 discusses how PROLOG can be used as a tool for answering questions about a knowledge base. Examples illustrating the concepts are provided.

Chapter 18 discusses some basic concepts in LISP and provides you with some understanding of the language.

Chapter 19 discusses a conceptual overview of neural networks.

Chapter 20 provides software examples of neural networks.

Also included is a bibliography that provides you with a list of books and articles, which can be used to augment this book, and a short description of the pertinent topics contained in each.

If you're a student, a programmer, a home computer novice, a professional using a personal computer in your business, a person already involved in artificial intelligence, or just curious about it, this book is for you.

A disc containing all the listed programs in the book is available. If you would like to order one, please see the back of this book for further information and instructions.

Robert I. Levine
Diane E. Drang
Barry Edelson

Acknowledgments

We would like to acknowledge Professor Robert Hong of Grumman Corporation and Polytechnic Institute of New York for his thought-provoking seminars in artificial intelligence.

This text is enriched by the considerable contribution of Sharon Juron Barlev in applying her extensive knowledge of Pascal to the field of artificial intelligence.

Contents

Preface	xii
Acknowledgments	xiv
Section 1 Human and Machine Intelligence	1
Chapter 1 An Overview of Intelligence	3
What is Artificial Intelligence?	3
What is Programming Like without Artificial Intelligence?	3
How Does Artificial Intelligence Make Programming Better?	4
How Does Human Intelligence Work?	4
Goals	4
Facts and Rules	5
Pruning	6
Inference Mechanism	8
Summary	8
Chapter 2 Developing an Artificial Intelligence System	10
Defining Goals	10
Defining Facts	12
Obtaining Data	13
Rules and Inferences	16
Verification through the Inference Mechanism	17
Pruning	19
Chapter 3 Defining Expert Systems	21
Heuristic Rules	22
Blackboard	23
You Can Do It	24
Chapter 4 Natural Language Processing	25
Lexical Analysis	27
Syntax Analysis	27
Semantic Analysis	28

Section 2	Inference Mechanisms: Tools for Machine Thinking	31
Chapter 5	Forward Chaining	33
	Example Using Forward Chaining	35
	The Knowledge Base	36
	Processing the Knowledge Base	37
	Forward Chaining Example	43
	Concepts for Design Implementation	46
	Programming Applications	47
	Forward Chaining Worksheet	53
Chapter 6	Backward Chaining	58
	A Procedure for Designing the Knowledge Base:	
	The Decision Tree	59
	Conversion to IF-THEN Rules	62
	Rule Generating Technique	64
	Processing the Knowledge Base	65
	Conclusion List	66
	Variable List	67
	Clause Variable List	68
	Conclusion Stack	70
	An Example Using the Knowledge Base	72
	Concepts for Design Implementation	79
	The Tool Itself	79
	Programming Applications	80
	Backward Chaining Worksheet	87
Chapter 7	Use of Probability and Fuzzy Logic in Expert Systems	92
	Fundamentals of Probability	93
	Bayesian Probability	94
	Example	95
	Fuzzy Concepts	97
	Probability Membership Table	98
	Summary	100
	Programming Applications	100
Section 3	Expert Systems: Knowledge Plus Inference	109
Chapter 8	Financial Planning Expert System	111
	How Do You Choose a Domain?	112
	How Do You Research Your Topic?	112
	Organizing the Relevant Facts for the Domain	113
	Decision Tree	117
	Backward and Forward Chaining Considerations	120
	Programming Applications	122

Chapter 9	Sales Expert System	124
	Establishing the Facts	125
	Salesperson Personality Types	125
	Instantiating the Facts	126
	Weighting Factors	127
	How Are Weighting Factors Used?	127
	An Example of the System at Work	127
	Assessing the Salesperson Personality Scores	128
	The Customer	130
	Assessing Customer Personality Scores	132
	Assessing the Possible Sales and Customer Combinations	133
	Expert System Assessments	134
	Programming Applications	135
Chapter 10	Learning Evaluation Expert System	140
	Organizing the Data into Topics and Subtopics	141
	Listing the Facts	142
	Assigning Weighting Factors and Establishing Decision Levels	143
	An Example of the System at Work	144
	Verification Using the Inference Mechanism	146
	Additional Applications	148
	Summary	152
	Programming Applications	152
Section 4	Advanced Programming Techniques for Powerful Systems	157
Chapter 11	Fundamentals of Object-Oriented Programming	159
	Creating a Structure	163
	Creating an Object	164
	Overview of Objects and Their Operations	165
	Operations on Objects	165
	Viewing Objects and Structures	166
	Object Operations	167
	Invoking Procedures	168
	A Method for Invoking Procedures	169
	Programming Applications	171
Chapter 12	Object-Oriented Programming: An Engineering Example	185
	Analog-to-Digital Conversion	185
	An Engineering System Using Object-Oriented Programming	188
	Structure	188
	Real-Time Data Acquisition	191
	Process Control	191
	Programming Applications	191

Chapter 13	Object-Oriented Expert Systems	199
	Designing the Structure	199
	Creating an Object	200
	Building the Knowledge Base	200
	Writing the Rules	201
	Using the Knowledge Base	203
	Backward or Forward Chaining	204
Section 5	Advanced Knowledge Representation for Smart Systems	205
Chapter 14	Semantic Nets	207
	Structure and Objects of Semantic Nets	207
	Rule-Based System Using Semantic Nets	208
	Programming Applications	209
Chapter 15	Certainty Factors	215
	Certainty Level Restrictions	217
	Programming Applications	217
Chapter 16	Automated Learning	221
	Example of a Learning System	222
	Programming Applications	225
Section 6	Languages Used in Artificial Intelligence	229
Chapter 17	Using PROLOG to Design Expert Systems	231
	Conceptual Example	231
	Review	234
	Converting Rules to PROLOG	234
	Summary	237
Chapter 18	LISP	238
	Introduction to LISP	238
	Function Evaluation	238
	Lists	240
	List Functions	241
	Predicates or Testing Functions	242
	Variable Assignments	243
	IF-THEN Rules through the Condition Function	244
	New Functions	245
	Summary	246

Section 7	Artificial Neural Networks	247
Chapter 19	Neural Networks Overview	249
	Neural Systems and Expert Systems	249
	The Unification of Learning Theory	250
	In the Beginning	250
	Design Philosophy of an Artificial Neural System	252
	Training the System	252
	Self-Modification	253
	Modeling the Neural System	254
	Modeling the Single Neuron	254
	Bias Voltage, Axon, Synapse, and Weighting Factor	255
	Summary	257
Chapter 20	Neural Networks—Software Design	258
	The Problem Statement	258
	Design of the Neural Network	259
	Initialization	259
	Training the Network	260
	Forward Propagation	261
	Applying the Input	261
	Calculating the Sum for Each Neuron	262
	Comparison with the Threshold	262
	Backward Propagation	263
	Is the Output Correct?	263
	Calculating the Error	263
	Changing the Weight Values	265
	Calculating the Blame	265
	Calculating the New Weight Values	266
	Is the Training Over?	267
	Programming Applications	267
	Other Applications—Image cognition	267
	Pixels and Image Recognition	267
	Multiple Levels of Neurons—Hidden Neurons	269
	Forward Propagation	269
	Backward Propagation	270
	Intermediate Error Determination	270
	Other Neural Network Configurations—Hebbian Networks	272
	Weight Matrix	273
	Calculating the Weights	274
	A Noisy Input	276
	Summary	277
	Bibliography and Recommended Readings	283
	Index	287

Human and Machine Intelligence

Section 1 introduces artificial intelligence and expert systems, acquainting us with how they work. Artificial intelligence is simply the transfer of intelligence to machines. Expert systems deal with a small area of expertise that can be converted from human to artificial intelligence.

We will take a close look at these concepts, using some common everyday experiences as examples. This will allow us to dissect the pieces of artificial intelligence and see how we can make them work for us. The process is simple, so let's get started.

An Overview of Intelligence

What Is Artificial Intelligence?

Artificial intelligence (AI) is simply a way of making a computer think intelligently. This is accomplished by studying how people think when they are trying to make decisions and solve problems, breaking those thought processes down into basic steps, and designing a computer program that solves problems using those same steps. AI thereby provides a simple, structured approach to designing complex decision-making programs.

What Is Programming Like without Artificial Intelligence?

A standard computer program can only provide answers to problems for which it is specifically programmed. If a standard program needs to be modified in order to accommodate new information, the entire program may have to be scanned until the optimum space is found to insert the modification. This is not only time consuming, but other parts of the program may be adversely affected in the process and errors may result.

Artificial intelligence, as its name implies, really does enable a computer to think. By simplifying the way programs are put together, AI imitates the basic human learning process by which new information is absorbed and made available for future reference. The human mind can incorporate new knowledge without changing the way the mind works or disturbing all the other facts that are already stored in the brain. An AI program works in very much the same way. It will become apparent as you read this book that changes made to AI pro-

grams are far simpler to implement than those made to standard programs.

How Does Artificial Intelligence Make Programming Better?

AI techniques allow the construction of a program in which each piece of the program represents a highly independent and identifiable step toward the solution of a problem or set of problems. Let's consider this carefully. Each piece of the program is like a piece of information in a person's mind. If that information is disputed, the mind can automatically adjust its thinking to accommodate a new set of facts. One doesn't have to go about reconsidering every piece of information one has ever learned, only those few pieces that are relevant to the particular change.

A standard program can do everything an artificial intelligence program can do, but it cannot be programmed as easily or as quickly. In both types of programs, all pieces are interdependent in the way they carry out their designed function. But an AI program possesses a notable characteristic which is equivalent to a vital characteristic of human intelligence. Each minute piece can be modified without affecting the structure of the entire program. This flexibility provides greater programming efficiency and understandability—in a word, intelligence.

How Does Human Intelligence Work?

Since AI is a science rooted in human thought processes, an examination of how people think is essential. Of course, no one knows exactly how the mind works. Human intelligence is a complex function that scientists have only begun to understand, but enough is known for us to make certain assumptions about how we think and to apply those assumptions in designing AI programs.

Goals

All thinking helps us accomplish something. When the alarm clock rings in the morning, a thought process must be employed to guide your hand to the button to turn it off. It isn't an automatic reaction; a specific response was sought to solve a particular problem. The final results to which all our thought processes are directed are called "goals."

Once you have reached the goal of turning off the alarm, your mind is immediately confronted with other goals to be reached, such as getting to the bathroom, brushing your teeth, getting dressed, making and eating breakfast, going to the bus stop, and so on. These are all goals which, when accomplished, lead you to the ultimate goal of getting to work and getting there on time. None of the thoughts guiding you to this final result are random or arbitrary. They have been pressed into service because every step of the way you had a specific goal in mind. When engaged in the most simple physical task or the most complex mental activity, the mind is sharply focused on a goal. Without goals, we have no reason to think. Examples of various types of goals are listed below:

1. Mapping the shortest route between New York and Boston
2. Deciding on the best type of wine to drink with certain fish
3. Learning to tie my shoes
4. Deciding how to determine if my child understands the concepts of arithmetic

When designing an AI system, the goal of the system must always be kept in mind. Remember, we don't do things because we think, we think because there are things we have to do.

Now that we understand where our minds are going when they're at work, let's consider just how we arrive at the multitude of goals we must reach every day.

Facts and Rules

We all know that the human mind possesses a vast store of knowledge relating to a countless array of objects and ideas. Survival depends on our ability to apply this knowledge to any situation that arises and to continuously learn from new experiences so that we will be able to respond to similar situations in the future. What is generally considered to be "intelligence" can be broken down into a collection of facts and a means of utilizing these facts to reach goals. This is done, in part, by formulating sets of rules relating to all the facts stored in the brain. An example of the type of facts and related rules we use every day follows:

Fact/rule set 1

Fact 1: A burning stove is hot.

Rule 1: If I put my hand on a burning stove, THEN it will hurt.