
This book gives an excellent overview of heuristic programs in the field of artificial intelligence. The author describes over 30 programs which "are doing things often called intelligent when done by humans." The book is divided into 13 chapters and each contains exercises and a bibliography.

Chapter 1 defines what the author means by "heuristic" since the book is only concerned with heuristic approaches to artificial intelligence problems. In addition, the author introduces the reader to "game trees." Chapter 2 describes programs which play chess, checkers, and kalah. The Greenblatt chess program, Samuel's checker player, and McCarthy's kalah program are described. Chapter 3 is also concerned with game-playing programs, namely five-in-a-row, quad, and card games. Chapter 4 describes programs that solve problems in chess, geometry, and calculus. Baylor and Simon's magic theorem algorithm is mentioned, but not given the space it richly deserves. Gelernter's geometry theorem prover and the author's symbolic automatic integrator (SAINt) are also described. Chapter 5 is concerned with automatic theorem proving, using Robinson's resolution principle. Wos' proof finder and Lee's consequence finder are described. Chapter 6 describes more heuristic programs that solve mathematical problems. A proof verifier, a heuristic regression-analysis program, a geometric analogy problem solver, a program that helps people manipulate mathematical expressions, a problem-solving program that plans, finds lemmas, and learns, and Newell, Shaw, and Simon's logic theorem are described. Chapter 7 is devoted to the author's multipurpose heuristic program (Multiple). Multiple used as a prover and kalah player is described. Chapter 8 describes the ambiguous work of Newell, Shaw, Simon, and Ernst with their general problem solver (GPS). The missionaries and cannibal problem, the tower of Hanoi problem, and other famous puzzle problems are mentioned. Chapter 9 describes some more multipurpose programs. Programs that balance assembly lines, write programs, compose music, and find chemical structures (heuristic Dendral) are briefly mentioned. Chapter 10 is concerned with automatic deductive question answering. Bobrow, Green, and Quillian's works are mentioned all too briefly. Chapter 11 reports on the author's work on automatically finding linear functions that make evaluations and recognize patterns. Chapter 12 describes Feigenbaum and Simon's elementary perceiver and memorizer (EPAM), and Chapter 13 gives the author's final words on the heuristic programming problem. This chapter is quite good and the author gives some valuable advice to heuristic programmers.

As the reader might guess, the author has written about many programs in only 196 pages and many sections suffer from the lack of detail. Also, the author has described his own work in greater depth than some comparable work by others which makes the book somewhat uneven in its treatment. Heuristic programs for solving the traveling salesman problem, for edge detection in pattern recognition, and for speech recognition have not been described. However, the book is intended to give the reader a sampling of the work taking place in artificial intelligence. The bibliography at the end of each chapter should be used to obtain greater detail for those areas of special interest to the reader. The exercises are sufficiently broad that they could be easily used to provoke classroom discussion. As a college text, the book coupled with some outside reading could give a fine introductory course to artificial intelligence.

DENNIS W. COOPER
Bell Tel. Lab., Inc.
Whippany, N. J. 07981


Goal and Result: The author states that the aim of this book is "1) to present an introduction to computers and programming using the System/370 as the primary vehicle; and 2) to present an introduction to the System/370 for people experienced in computer science." I have found that this book contains a lot of useful information and, in fact, I can say that I learned quite a bit from the book. However, this does not mean that the author has met his goals. As for the beginner, he would not come out with a thorough understanding of S/370 or computers in general because so little space was devoted to his initiation. The intermediate or advanced computer scientist would learn much about S/370 but I am afraid he would be left with many questions about the beast, just as I was.

Divisions: The author divides the book into three parts: 1) computer and programming concepts; 2) the structure of System/370; and 3) System/370 systems and devices. My review will examine these three parts in that order.

1) Concepts: The presentation of concepts here is at too simple a level for anyone who has a background in computers. On the other hand, it is not thorough enough for the beginner. Let me present a few examples. The author describes binary numbers, but he does not explain the binary system of counting. He says that the word may hold a mixed fraction (e.g., 12.6) but he does not explain how this is done—i.e., really is not done in hardware—the assembler or compiler provides the machinery. As a final example, he explains floating point but uses decimal numbers to illustrate. This will only confuse the beginner and is unnecessary to the person with any background.

2) Architecture: This section summarizes information provided in IBM Principles of Operations Handbooks. In presenting this information the author must be in a quandary about what should be omitted versus what should be duplicated from the manuals. This is a difficult problem and more often than not I disagree with the selection that the author has taken. Some examples may clarify what I mean.

The author mentions that negative numbers are represented in 2's complement form. He gives an example of this showing how minus 5 is represented but he does not explain the principle.

The author discusses the contents of the program status word, labeling all the fields that it contains. He explains the use of some of these but does not explain others, thus creating curiosity in the reader.

A similar comment can be made about the control registers whose contents are enumerated. Now there is a mystery: who sets these registers; who references them; when and why are they used?
My most serious objection is that there is no distinction for contents and function for System/360 in contrast to System/370. Several tables enumerate the instructions set for System/370. Similar tables may be found in the Principles of Operation. Only a token description is provided. Can you imagine the complex command being allocated a single sentence?

3) Systems and Devices: I found this part most useful. Here the specific features of each computer model are described. There is enough detail provided to give the principles of the new features. I would like to see detail at the register level, but this is strictly my own prejudice.

There is a particularly good discussion of buffer storage and its control. It makes clear how buffer storage expedites the data and instructions that have been recently referenced, or where immediate neighbors have been recently referenced.

There is only a token description of I/O devices. My curiosity about the 3330 disk system was not satisfied.

Ivan Flores
Baruch College
City Univ. New York
New York, N. Y. 10010

B72-27 The Fourth Generation—International Computer State of the Art Report/#1. (Maiden Head, Berks., England: Infotech, Limited, 501 pp., $95.00, subscriptions available for series of reports to be issued.)

The book is composed of three major parts: a dialogue on the fourth generation theme, a series of papers (called presentations), and finally six invited papers.

Of the three parts the “presentations” part is by far the best. The dialogue presented in the first part is constructed by the editors by excerpting the papers. This editorial device has recently become very fashionable; however, it is very difficult for the editors to excerpt avoiding distortions and nonsequiturs. The result is that this "cute" technique does not lend itself to expressing serious and insightful thought.

The “Presentations” part is composed of the following twelve papers.


"Fourth Generation Computer Design," by G. G. Scarrott. Gives a uniquely insightful picture of the evolution of computing systems and the key role software and hardware interfaces play in keeping the system complexity to a human scale. Interfaces such as the channel level interface allow organizationally separate groups to develop a portion of the system such as the main frame I/O controller and a peripheral processor in a relatively independent mode.

"Addressing Structures: The Focal Point of Fourth Generation Architecture," by N. D. Gammage. Focuses correctly on one of the central architectural issues of the fourth generation and does so with a very lucid discussion of the notions underlying paging, segmentation, and codeword systems.


"Machine-Independent Operating Systems: A Functional Approach to Design," by P. Cox. A number of somewhat trivial points are made (e.g., OS should be clean and tight) with excessive use of jargon.

"Time Sharing Management Software: Man & Machine," by D. F. Hartley. Discusses in an interesting manner a very neglected and important area, i.e., the management of the installation as a complement to system design.

"The Nature & Benefits of Modular Operating Systems," by D. Morris. Gives the reader a good understanding of the underlying fundamental issues of a process oriented OS while presenting a design pathway at the University of Manchester. In spite of the fact that some aspects of this specific design are questionable, reading the paper is a very valuable experience.


"Network Systems: The Economic Solution for the Fourth Generation User," by I. M. Barron. A set of arguments (mostly economic and marketing) why IBM will not release a fourth generation line of systems and why the rest of the industry would not either.


"The Interaction of Software Engineering & Machine Structure," by C. Strachey. Use of high-level languages is mandatory for development efficiency. Thus we must learn empirically to optimize the machine architecture for support of the chosen high-level language.

The third and final part of the book contains the following invited papers.

"Concern for Correctness as a Guiding Principle for Program Composition," by E. W. Dijkstra. Basic theme is: we have to learn how to put structure in programs so their correctness can be established logically.

"A Survey of Store Management Techniques," by C. A. R. Hoare and R. M. McKeag. Surveys a number of page replacement algorithms with no mention of the role of LRU and OPT (min) algorithms.


"Macros and Their Uses," by P. J. Brown. A tutorial on macros and their use. It addresses a very neglected area. Advocates a general purpose macro preprocessor for a fourth generation system.


"Manageable Software Engineering," by R. W. Bemer. Several interesting notions concerning the management of software development. Stresses the importance of suitable development environment (quick turn-around facility).

U. O. Gagliardi
Harvard Univ.
Cambridge, Mass.