Artificial Intelligence Using C

Herbert Schildt

Artificial Intelligence Using C

Herbert Schildt

Osborne **McGraw-Hill**
Berkeley, California

Osborne **McGraw-Hill**
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations and book distributors outside of the U.S.A., please write to Osborne **McGraw-Hill** at the above address.

A complete list of trademarks appears on page 403.

Artificial Intelligence Using C

1234567890 DODO 8987

ISBN 0-07-881255-0

PREFACE

Few if any of the events that have occurred or will occur in the last quarter of this century will have as profound or lasting an effect upon human life as the creation of intelligent machines. The introduction of smart, autonomous computers and robots will cause a fundamental restructuring of society. To understand why this will happen requires that you understand two important premises. The first premise is that virtually all current uses and applications of computers and automation in industry are direct extensions of the basic principles of the industrial revolution: specifically, the use of computers and automation has replaced workers who perform low-level, repetitive tasks. The second premise is that the introduction of *smart automation* will cause a second industrial revolution. However, in this revolution, the workers who will be replaced by machines will be from middle management, which comprises those jobs that require cognitive decision-making (but, perhaps, not insight or inventiveness).

To see the effect of smart automation upon the basic corporate structure, think of a company as a pyramid, as shown on the next page: the president is at the top, with the high-level executives at the next level below, then layers of middle management, and finally the lower levels of semi- and unskilled workers.

Disk Order Form

Please send me ____ copies, at $24.95 each, of the listings in *Artificial Intelligence Using C. Foreign orders:* add $5.00 for shipping and handling.

____
Name

____
Address

____
City State ZIP

____
Telephone

Method of payment: Check____ VISA____ MC____

Credit card number: ______________________________

Expiration date: ______________________________

Signature: ______________________________

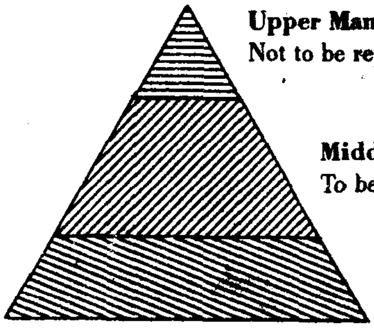
Send to: Herbert Schildt
RR 1, Box 130
Mahomet, IL 61853

Chapter 1 begins with a brief history of AI, and then discusses exactly what artificial intelligence means and implies. Chapter 2 deals with problem solving, while Chapter 3 focuses on expert systems. Chapter 4 moves on to natural-language processing. Next are vision, robotics, and machine learning in Chapters 5, 6, and 7, respectively. The final chapters deal with comparing logic and uncertainty, and the ways to make a computer appear to be human.

In addition to illustrating various concepts, this book provides applications that can serve as starting points from which you can develop complete systems. For example, you can easily expand and enhance the expert system that is developed in Chapter 3 or the natural-language parsers of Chapter 4.

As you probably know, keying in listings from a book can be a problem. You can easily make mistakes or accidentally omit a line. The process can be a time-consuming and frustrating experience. For these reasons, I am offering an IBM PC-compatible diskette that contains all of the listings in this book. The cost of the diskette is $24.95. You may obtain it by mailing your order and payment with the form that follows. If you are in a hurry, you may call (217) 586-4021 to place your order.

—H.S.



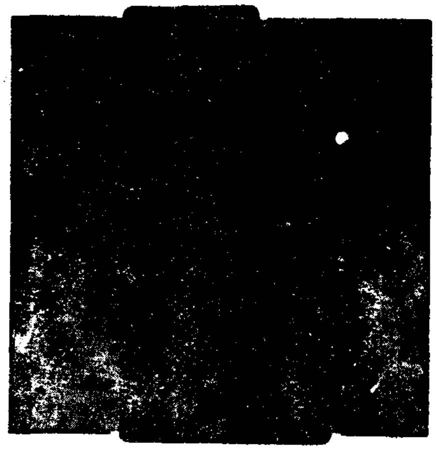
Simple automation has already replaced many blue-collar workers; thus, in the next step, smart automation will replace the mid-section of the pyramid. It is now an understatement to say that this restructuring of the workforce will have significant impact upon society. However, consider the fact that having this discussion would have seemed far-fetched, say, 10 years ago when hardly any artificial intelligence (AI) applications existed. In fact, most computer science majors today have never taken even a single course in AI.

The purpose of this book is to give you as a programmer the necessary background in the central topics and issues of AI, and some of the basic techniques that you can use to address them. While the field of AI is large, the central topics are

- expert systems
- problem solving
- logic and uncertainty
- natural-language processing
- robotics
- machine learning
- vision and pattern recognition
- real-world interfacing

CONTENTS

CHAPTER



Artificial Intelligence: A Quick Overview

Until recently, many people have viewed the field of artificial intelligence, or AI, as the darker side of computer science: they believed that, like Shelley's Dr. Frankenstein attempting to create life, the AI programmers worked to create thought. Artificial intelligence researchers were, at times, paradoxically regarded as both the elite and the "lunatic fringe" of computer science. When forced to give a view on the feasibility or practicality of machine intelligence, mainstream wage-earning programmers, usually careful to avoid references to AI, would always say that "much research still needs to be done," and that "sometime in the distant future, there will be important discoveries, but so far the field has produced precious little." Views on artificial intelligence have definitely changed!

In less than five years, artificial intelligence has gone from being a small backwater of computer science to being the hottest thing to happen to computers since perhaps the transistor! This rapid change is based on four main factors: the success of expert systems, which were the first truly financially successful AI products; the well-publicized commitment of the Japanese to AI; the slow but steady integration of AI techniques into existing applications; and finally, the fact that AI's time has come.

In this book, you will explore the various topics and techniques that compose the field of artificial intelligence. The most important and most common of these are the subjects of the remaining chapters in this book. However, before you move on to these applications, this chapter will review the history of AI and then determine whether or not it is possible for a computer to be intelligent.

A SHORT HISTORY OF AI

It is difficult to pinpoint an exact starting date for what is commonly called AI. Perhaps credit for the birth of AI should be given to A. M. Turing for his invention of the stored program computer. The first computers were actually dedicated machines that literally had to be rewired to solve different problems. Turing's recognition that programs could be stored as data in the computer's memory and executed later formed the basis for all modern computers. The storing of programs allowed the computer to change its function quickly and easily by simply running a new program. This capability implies that a computer *might* be able to change its *own* function—that is, to learn or think!

However, what is commonly thought of as AI began around 1960 when, at MIT, John McCarthy created LISP—the first artificial intelligence research language. The term *artificial intelligence* is generally credited to Marvin Minsky, also of MIT, who in 1961 wrote a paper entitled "Steps towards Artificial Intelligence" (*The Institute of Radio Engineers Proceedings* 49 [January 1961]). The 1960s were a period of intense optimism over the possibility of making a computer think. After all, the 1960s saw the first chess-playing computer, the first computerized mathematical proofs, and the famous—and highly publicized—ELIZA program, which was written in 1964 by Joseph Weizenbaum of MIT. The ELIZA program acted like a Rogerian psychoanalyst. In this style of analysis, the psychiatrist takes a passive role, generally echoing the patient's own remarks, instead of leading the conversation—just the type of task a computer can easily do. At the time, the program caused quite a stir. People asked, "Should computers be used in this way? Can the computerized psychiatrist actually be better than a human one? Should this type of program be allowed?" Even Weizenbaum, ELIZA's creator, wrote the book *Computer Power and Human Reason* (San Francisco: W. H. Freeman & Co., 1976) essentially to discredit his own program! (Remember that the 1960s were also a period of intense *fear* of automation. It is sometimes hard to understand the emotions that existed just over 20 years ago.)

Because of the apparent successes of AI, it seemed that the goal of producing a program that had humanlike intelligence was just around the corner. However, this was not to be so.

What was not clearly understood in the 1960s was the difficulty of generalizing these specific successes into a flexible, intelligent program. This book will examine some of the reasons for this. As programmers attempted to increase the generality of certain programs, these attempts required greater computing resources than those that were currently available. Thus, either memory was quickly exhausted or execution time became too slow.

By the mid 1970s, computers with large memories were common and computing speeds had increased dramatically. However, even with these improvements, many of the older approaches to AI continued to fail because there was inherent inefficiency. For example, consider something as simple as sorting an array of numbers. If you use a bubble sort, the sort time will be proportional to N^2, where N is the number of elements. This means that if sorting a 10-element array takes 1 second, then sorting a 100-element array will take 100 seconds, and so on. You can see that, at some point, there is an array that will take longer to sort than the average person lives! No matter how fast the computer becomes, a sorting algorithm whose time is proportional to N^2 will quickly become too slow. A better approach is to alter the sorting algorithm so that it is more efficient. For example, the sorting time of the QuickSort algorithm is proportional to $N^{1.2}$—a significant improvement. Figure 1-1 shows the difference between the curves of the bubble sort and the QuickSort. Just as new sorting algorithms were needed to improve sorting times, many AI problems required improved routines, which enabled programs to execute quickly enough or to make better use of memory, to allow those problems to be solved.

By the end of the 1970s several successes—such as natural-language processing, knowledge representation, and problem solving—had been achieved in specific areas of AI. These successes set the stage for the introduction of the first commercial AI product, the *expert system*. An expert system is a program that contains information about a certain field and, when interrogated, responds much like a human expert. One of the first expert systems was MYCIN, which was developed at Stanford University to help physicians diagnose illnesses.

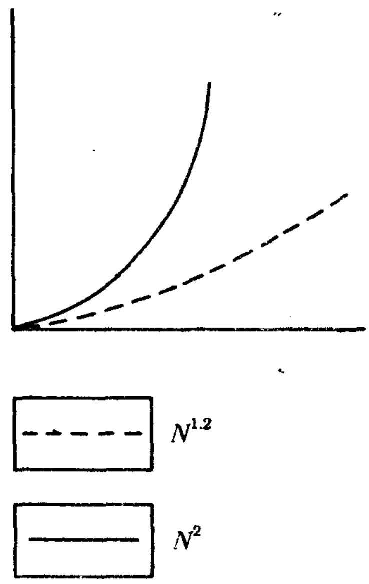


Figure 1-1. A comparison of an N^2 curve and an $N^{1.2}$ curve

One of the most important AI events that occurred in the 1970s was virtually overlooked in the United States until the 1980s. This event was the creation of Prolog in 1972 by Alain Colmerauer in Marseilles, France. Like LISP, Prolog was a language that was designed to help solve AI-related problems; unlike LISP, it had several special features, such as a built-in database, and it had a rather simple syntax. Essentially, by 1980, LISP was the AI language of choice in the U.S., while Prolog had the same status in Europe. However, in 1981, this situation changed with the announcement by the Japanese that they would be using Prolog as the basis for their much-touted "fifth-generation" computers. American programmers were caught off guard!

What makes Prolog so important in the history of AI is that it embodied a deeper understanding of the thought process than LISP did. For example, Prolog contains a built-in database facility and backtracking routines, both of which are necessary for many problem-solving situations. (In fact, as you will see, the backtracking support needs to be coded explicitly in C in order to implement many AI techniques.) Although Prolog has been gaining popularity in the United States since 1981, it is unclear whether or not it will become the premier AI research language in the U.S.

Presently, the emphasis in the field of AI is shifting from research to implementation. This shift means that the AI techniques that were developed in the laboratory by using a research language will need to be implemented by using various general-purpose languages to solve real-world applications. As this book will show you, this task is not as formidable as you might think.

CAN COMPUTERS THINK?

Before you can explore the realm of AI through C, you must understand what it means for a computer to think. The concept of a thinking computer implies that the computer is executing a thinking program. For this discussion and in order to remain compatible with traditional terms, this book will refer to a thinking program as an *intelligent program*. However, there is much debate about whether or not intelligent programs —and, hence, thinking computers —even exist! This debate is not easy to settle because it is decided, in part, by the way that the definition of intelligence is interpreted. There are compelling (and sometimes emotional) arguments to support each conclusion. A question that comes up in this debate is how an intelligent program differs from a "non-intelligent" program. This section explores a few of these arguments — however, you will have to decide the outcome for yourself.

Determining what an intelligent program is implies that one know what intelligence means. One dictionary defines *intelligence* as "the capacity to apprehend facts and propositions and their relations, and to reason about them." This definition leads to the question, "What does *reason* mean?" In this context, it means to *think* — which is where the trouble lies! Long ago, it was suggested that people cannot explain *how* they think, but they can tell *what* they think! The fact is that people cannot really understand how they think. (If they did, it might not be such a big job to make a computer do it!)

If you stay with a strict interpretation of the definition of intelligence, then you could argue that *all* programs are intelligent. Consider this: The first part of the definition of intelligence is the ability to apprehend facts, propositions, and relations. Computers are amazingly well-suited to performing these types of tasks. For example, a relational database can store (apprehend) information, accept queries (propositions), and, as its name implies, represent relations. Certainly, some types of information, such as visual images, are much harder for a computer to apprehend than others, but the definition of intelligence does not demand that the apprehension take a specific form — it only demands that the apprehension take place. Therefore, what computers normally do — gather, store, and access information — satisfies the first requirement for intelligence.

However, can the database *reason* with these facts — which is the second requirement for intelligence? Perhaps. The answer depends upon what you will accept as the definition of *reason*. If the database's manipulation of information — the act of searching, sorting, query processing, filing, and the like — can be called reasoning, then one must say that the database is an intelligent program. This implies that most computer programs are intelligent. Remember that most computer programs manipulate information in reasonable, logical ways. Therefore, this form of reasoning must qualify as intelligence.