1 INTRODUCTION

In which we try to explain why we consider artificial intelligence to be a subject most worthy of study, and in which we try to decide what exactly it is, this being a good thing to decide before embarking.

ARTIFICIAL INTELLIGENCE

Humankind has given itself the scientific name **homo sapiens**—man the wise—because our mental capacities are so important to our everyday lives and our sense of self. The field of **artificial intelligence**, or AI, attempts to understand intelligent entities. Thus, one reason to study it is to learn more about ourselves. But unlike philosophy and psychology, which are also concerned with intelligence, AI strives to *build* intelligent entities as well as understand them. Another reason to study AI is that these constructed intelligent entities are interesting and useful in their own right. AI has produced many significant and impressive products even at this early stage in its development. Although no one can predict the future in detail, it is clear that computers with human-level intelligence (or better) would have a huge impact on our everyday lives and on the future course of civilization.

AI addresses one of the ultimate puzzles. How is it possible for a slow, tiny brain, whether biological or electronic, to perceive, understand, predict, and manipulate a world far larger and more complicated than itself? How do we go about making something with those properties? These are hard questions, but unlike the search for faster-than-light travel or an antigravity device, the researcher in AI has solid evidence that the quest is possible. All the researcher has to do is look in the mirror to see an example of an intelligent system.

AI is one of the newest disciplines. It was formally initiated in 1956, when the name was coined, although at that point work had been under way for about five years. Along with modern genetics, it is regularly cited as the "field I would most like to be in" by scientists in other disciplines. A student in physics might reasonably feel that all the good ideas have already been taken by Galileo, Newton, Einstein, and the rest, and that it takes many years of study before one can contribute new ideas. AI, on the other hand, still has openings for a full-time Einstein.

The study of intelligence is also one of the oldest disciplines. For over 2000 years, philosophers have tried to understand how seeing, learning, remembering, and reasoning could, or should, be done.[1] The advent of usable computers in the early 1950s turned the learned but armchair speculation concerning these mental faculties into a real experimental and theoretical discipline. Many felt that the new "Electronic Super-Brains" had unlimited potential for intelligence. "Faster Than Einstein" was a typical headline. But as well as providing a vehicle for creating artificially intelligent entities, the computer provides a tool for testing theories of intelligence, and many theories failed to withstand the test—a case of "out of the armchair, into the fire." AI has turned out to be more difficult than many at first imagined, and modern ideas are much richer, more subtle, and more interesting as a result.

AI currently encompasses a huge variety of subfields, from general-purpose areas such as perception and logical reasoning, to specific tasks such as playing chess, proving mathematical theorems, writing poetry, and diagnosing diseases. Often, scientists in other fields move gradually into artificial intelligence, where they find the tools and vocabulary to systematize and automate the intellectual tasks on which they have been working all their lives. Similarly, workers in AI can choose to apply their methods to any area of human intellectual endeavor. In this sense, it is truly a universal field.

1.1 WHAT IS AI?

We have now explained why AI is exciting, but we have not said what it *is*. We could just say, "Well, it has to do with smart programs, so let's get on and write some." But the history of science shows that it is helpful to aim at the right goals. Early alchemists, looking for a potion for eternal life and a method to turn lead into gold, were probably off on the wrong foot. Only when the aim changed, to that of finding explicit theories that gave accurate predictions of the terrestrial world, in the same way that early astronomy predicted the apparent motions of the stars and planets, could the scientific method emerge and productive science take place.

Definitions of artificial intelligence according to eight recent textbooks are shown in Figure 1.1. These definitions vary along two main dimensions. The ones on top are concerned with *thought processes* and *reasoning*, whereas the ones on the bottom address *behavior*. Also, the definitions on the left measure success in terms of *human* performance, whereas the ones

RATIONALITY

on the right measure against an *ideal* concept of intelligence, which we will call **rationality**. A system is rational if it does the right thing. This gives us four possible goals to pursue in artificial intelligence, as seen in the caption of Figure 1.1.

Historically, all four approaches have been followed. As one might expect, a tension exists between approaches centered around humans and approaches centered around rationality.[2] A human-centered approach must be an empirical science, involving hypothesis and experimental

[1] A more recent branch of philosophy is concerned with proving that AI is impossible. We will return to this interesting viewpoint in Chapter 26.

[2] We should point out that by distinguishing between *human* and *rational* behavior, we are not suggesting that humans are necessarily "irrational" in the sense of "emotionally unstable" or "insane." One merely need note that we often make mistakes; we are not all chess grandmasters even though we may know all the rules of chess; and unfortunately, not everyone gets an A on the exam. Some systematic errors in human reasoning are cataloged by Kahneman *et al.* (1982).

"The exciting new effort to make computers think ... *machines with minds*, in the full and literal sense" (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)	"The study of mental faculties through the use of computational models" (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act" (Winston, 1992)
"The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)	"A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990) "The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)

Figure 1.1 Some definitions of AI. They are organized into four categories:

Systems that think like humans.	Systems that think rationally.
Systems that act like humans.	Systems that act rationally.

confirmation. A rationalist approach involves a combination of mathematics and engineering. People in each group sometimes cast aspersions on work done in the other groups, but the truth is that each direction has yielded valuable insights. Let us look at each in more detail.

Acting humanly: The Turing Test approach

TURING TEST

The **Turing Test**, proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence. Turing defined intelligent behavior as the ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator. Roughly speaking, the test he proposed is that the computer should be interrogated by a human via a teletype, and passes the test if the interrogator cannot tell if there is a computer or a human at the other end. Chapter 26 discusses the details of the test, and whether or not a computer is really intelligent if it passes. For now, programming a computer to pass the test provides plenty to work on. The computer would need to possess the following capabilities:

NATURAL LANGUAGE PROCESSING

◇ **natural language processing** to enable it to communicate successfully in English (or some other human language);

KNOWLEDGE REPRESENTATION

◇ **knowledge representation** to store information provided before or during the interrogation;

AUTOMATED REASONING

◇ **automated reasoning** to use the stored information to answer questions and to draw new conclusions;

MACHINE LEARNING

◇ **machine learning** to adapt to new circumstances and to detect and extrapolate patterns.

Turing's test deliberately avoided direct physical interaction between the interrogator and the computer, because *physical* simulation of a person is unnecessary for intelligence. However,

TOTAL TURING TEST

the so-called **total Turing Test** includes a video signal so that the interrogator can test the subject's perceptual abilities, as well as the opportunity for the interrogator to pass physical objects "through the hatch." To pass the total Turing Test, the computer will need

COMPUTER VISION

ROBOTICS

- ◊ **computer vision** to perceive objects, and
- ◊ **robotics** to move them about.

Within AI, there has not been a big effort to try to pass the Turing test. The issue of acting like a human comes up primarily when AI programs have to interact with people, as when an expert system explains how it came to its diagnosis, or a natural language processing system has a dialogue with a user. These programs must behave according to certain normal conventions of human interaction in order to make themselves understood. The underlying representation and reasoning in such a system may or may not be based on a human model.

Thinking humanly: The cognitive modelling approach

COGNITIVE SCIENCE

If we are going to say that a given program thinks like a human, we must have some way of determining how humans think. We need to get *inside* the actual workings of human minds. There are two ways to do this: through introspection—trying to catch our own thoughts as they go by—or through psychological experiments. Once we have a sufficiently precise theory of the mind, it becomes possible to express the theory as a computer program. If the program's input/output and timing behavior matches human behavior, that is evidence that some of the program's mechanisms may also be operating in humans. For example, Newell and Simon, who developed GPS, the "General Problem Solver" (Newell and Simon, 1961), were not content to have their program correctly solve problems. They were more concerned with comparing the trace of its reasoning steps to traces of human subjects solving the same problems. This is in contrast to other researchers of the same time (such as Wang (1960)), who were concerned with getting the right answers regardless of how humans might do it. The interdisciplinary field of **cognitive science** brings together computer models from AI and experimental techniques from psychology to try to construct precise and testable theories of the workings of the human mind.

Although cognitive science is a fascinating field in itself, we are not going to be discussing it all that much in this book. We will occasionally comment on similarities or differences between AI techniques and human cognition. Real cognitive science, however, is necessarily based on experimental investigation of actual humans or animals, and we assume that the reader only has access to a computer for experimentation. We will simply note that AI and cognitive science continue to fertilize each other, especially in the areas of vision, natural language, and learning. The history of psychological theories of cognition is briefly covered on page 12.

Thinking rationally: The laws of thought approach

SYLLOGISMS

The Greek philosopher Aristotle was one of the first to attempt to codify "right thinking," that is, irrefutable reasoning processes. His famous **syllogisms** provided patterns for argument structures that always gave correct conclusions given correct premises. For example, "Socrates is a man;

all men are mortal; therefore Socrates is mortal." These laws of thought were supposed to govern

LOGIC

the operation of the mind, and initiated the field of **logic**.

The development of formal logic in the late nineteenth and early twentieth centuries, which we describe in more detail in Chapter 6, provided a precise notation for statements about all kinds of things in the world and the relations between them. (Contrast this with ordinary arithmetic notation, which provides mainly for equality and inequality statements about numbers.) By 1965, programs existed that could, given enough time and memory, take a description of a problem in logical notation and find the solution to the problem, if one exists. (If there is no solution,

LOGICIST

the program might never stop looking for it.) The so-called **logicist** tradition within artificial intelligence hopes to build on such programs to create intelligent systems.

There are two main obstacles to this approach. First, it is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain. Second, there is a big difference between being able to solve a problem "in principle" and doing so in practice. Even problems with just a few dozen facts can exhaust the computational resources of any computer unless it has some guidance as to which reasoning steps to try first. Although both of these obstacles apply to *any* attempt to build computational reasoning systems, they appeared first in the logicist tradition because the power of the representation and reasoning systems are well-defined and fairly well understood.

Acting rationally: The rational agent approach

AGENT

Acting rationally means acting so as to achieve one's goals, given one's beliefs. An **agent** is just something that perceives and acts. (This may be an unusual use of the word, but you will get used to it.) In this approach, AI is viewed as the study and construction of rational agents.

In the "laws of thought" approach to AI, the whole emphasis was on correct inferences. Making correct inferences is sometimes *part* of being a rational agent, because one way to act rationally is to reason logically to the conclusion that a given action will achieve one's goals, and then to act on that conclusion. On the other hand, correct inference is not *all* of rationality, because there are often situations where there is no provably correct thing to do, yet something must still be done. There are also ways of acting rationally that cannot be reasonably said to involve inference. For example, pulling one's hand off of a hot stove is a reflex action that is more successful than a slower action taken after careful deliberation.

All the "cognitive skills" needed for the Turing Test are there to allow rational actions. Thus, we need the ability to represent knowledge and reason with it because this enables us to reach good decisions in a wide variety of situations. We need to be able to generate comprehensible sentences in natural language because saying those sentences helps us get by in a complex society. We need learning not just for erudition, but because having a better idea of how the world works enables us to generate more effective strategies for dealing with it. We need visual perception not just because seeing is fun, but in order to get a better idea of what an action might achieve—for example, being able to see a tasty morsel helps one to move toward it.

The study of AI as rational agent design therefore has two advantages. First, it is more general than the "laws of thought" approach, because correct inference is only a useful mechanism for achieving rationality, and not a necessary one. Second, it is more amenable to scientific development than approaches based on human behavior or human thought, because the standard of rationality is clearly defined and completely general. Human behavior, on the other hand, is well-adapted for one specific environment and is the product, in part, of a complicated and largely unknown evolutionary process that still may be far from achieving perfection. *This book will therefore concentrate on general principles of rational agents, and on components for constructing them.* We will see that despite the apparent simplicity with which the problem can be stated, an enormous variety of issues come up when we try to solve it. Chapter 2 outlines some of these issues in more detail.

One important point to keep in mind: we will see before too long that achieving perfect rationality—always doing the right thing—is not possible in complicated environments. The computational demands are just too high. However, for most of the book, we will adopt the working hypothesis that understanding perfect decision making is a good place to start. It simplifies the problem and provides the appropriate setting for most of the foundational material in the field. Chapters 5 and 17 deal explicitly with the issue of **limited rationality**—acting appropriately when there is not enough time to do all the computations one might like.

LIMITED RATIONALITY

1.2 THE FOUNDATIONS OF ARTIFICIAL INTELLIGENCE

In this section and the next, we provide a brief history of AI. Although AI itself is a young field, it has inherited many ideas, viewpoints, and techniques from other disciplines. From over 2000 years of tradition in philosophy, theories of reasoning and learning have emerged, along with the viewpoint that the mind is constituted by the operation of a physical system. From over 400 years of mathematics, we have formal theories of logic, probability, decision making, and computation. From psychology, we have the tools with which to investigate the human mind, and a scientific language within which to express the resulting theories. From linguistics, we have theories of the structure and meaning of language. Finally, from computer science, we have the tools with which to make AI a reality.

Like any history, this one is forced to concentrate on a small number of people and events, and ignore others that were also important. We choose to arrange events to tell the story of how the various intellectual components of modern AI came into being. We certainly would not wish to give the impression, however, that the disciplines from which the components came have all been working toward AI as their ultimate fruition.

Philosophy (428 B.C.–present)

> The safest characterization of the European philosophical tradition is that it consists of a series of footnotes to Plato.
> —Alfred North Whitehead

We begin with the birth of Plato in 428 B.C. His writings range across politics, mathematics, physics, astronomy, and several branches of philosophy. Together, Plato, his teacher Socrates, and his student Aristotle laid the foundation for much of western thought and culture. The philosopher Hubert Dreyfus (1979, p. 67) says that "The story of artificial intelligence might well begin around 450 B.C." when Plato reported a dialogue in which Socrates asks Euthyphro,[3] "I want to know what is characteristic of piety which makes all actions pious ... that I may have it to turn to, and to use as a standard whereby to judge your actions and those of other men."[4] In other words, Socrates was asking for an *algorithm* to distinguish piety from non-piety. Aristotle went on to try to formulate more precisely the laws governing the rational part of the mind. He developed an informal system of syllogisms for proper reasoning, which in principle allowed one to mechanically generate conclusions, given initial premises. Aristotle did not believe all parts of the mind were governed by logical processes; he also had a notion of intuitive reason.

Now that we have the idea of a set of rules that can describe the working of (at least part of) the mind, the next step is to consider the mind as a physical system. We have to wait for René Descartes (1596–1650) for a clear discussion of the distinction between mind and matter, and the problems that arise. One problem with a purely physical conception of the mind is that it seems to leave little room for free will: if the mind is governed entirely by physical laws, then it has no more free will than a rock "deciding" to fall toward the center of the earth. Although a

DUALISM

strong advocate of the power of reasoning, Descartes was also a proponent of **dualism**. He held that there is a part of the mind (or soul or spirit) that is outside of nature, exempt from physical laws. On the other hand, he felt that animals did not possess this dualist quality; they could be considered as if they were machines.

MATERIALISM

An alternative to dualism is **materialism**, which holds that all the world (including the brain and mind) operate according to physical law.[5] Wilhelm Leibniz (1646–1716) was probably the first to take the materialist position to its logical conclusion and build a mechanical device intended to carry out mental operations. Unfortunately, his formulation of logic was so weak that his mechanical concept generator could not produce interesting results.

It is also possible to adopt an intermediate position, in which one accepts that the mind has a physical basis, but denies that it can be *explained* by a reduction to ordinary physical processes. Mental processes and consciousness are therefore part of the physical world, but inherently unknowable; they are beyond rational understanding. Some philosophers critical of AI have adopted exactly this position, as we discuss in Chapter 26.

Barring these possible objections to the aims of AI, philosophy had thus established a tradition in which the mind was conceived of as a physical device operating principally by reasoning with the knowledge that it contained. The next problem is then to establish the

EMPIRICIST

source of knowledge. The **empiricist** movement, starting with Francis Bacon's (1561–1626) *Novum Organum*,[6] is characterized by the dictum of John Locke (1632–1704): "Nothing is in the understanding, which was not first in the senses." David Hume's (1711–1776) *A Treatise*

INDUCTION

of Human Nature (Hume, 1978) proposed what is now known as the principle of **induction**:

---

3 The *Euthyphro* describes the events just before the trial of Socrates in 399 B.C. Dreyfus has clearly erred in placing it 51 years earlier.

4 Note that other translations have "goodness/good" instead of "piety/pious."

5 In this view, the perception of "free will" arises because the deterministic generation of behavior is constituted by the operation of the mind selecting among what appear to be the possible courses of action. They remain "possible" because the brain does not have access to its own future states.

6 An update of Aristotle's *organon*, or instrument of thought.

that general rules are acquired by exposure to repeated associations between their elements. The theory was given more formal shape by Bertrand Russell (1872–1970) who introduced **logical positivism**. This doctrine holds that all knowledge can be characterized by logical theories connected, ultimately, to **observation sentences** that correspond to sensory inputs.[7] The **confirmation theory** of Rudolf Carnap and Carl Hempel attempted to establish the nature of the connection between the observation sentences and the more general theories—in other words, to understand how knowledge can be acquired from experience.

LOGICAL POSITIVISM

OBSERVATION SENTENCES

CONFIRMATION THEORY

The final element in the philosophical picture of the mind is the connection between knowledge and action. What form should this connection take, and how can particular actions be justified? These questions are vital to AI, because only by understanding how actions are justified can we understand how to build an agent whose actions are justifiable, or rational. Aristotle provides an elegant answer in the *Nicomachean Ethics* (Book III. 3, 1112b):

> We deliberate not about ends, but about means. For a doctor does not deliberate whether he shall heal, nor an orator whether he shall persuade, nor a statesman whether he shall produce law and order, nor does any one else deliberate about his end. They assume the end and consider how and by what means it is attained, and if it seems easily and best produced thereby; while if it is achieved by one means only they consider *how* it will be achieved by this and by what means *this* will be achieved, till they come to the first cause, which in the order of discovery is last . . . and what is last in the order of analysis seems to be first in the order of becoming. And if we come on an impossibility, we give up the search, e.g. if we need money and this cannot be got; but if a thing appears possible we try to do it.

Aristotle's approach (with a few minor refinements) was implemented 2300 years later by Newell and Simon in their GPS program, about which they write (Newell and Simon, 1972):

MEANS–ENDS ANALYSIS

> The main methods of GPS jointly embody the heuristic of **means–ends analysis**. Means–ends analysis is typified by the following kind of common-sense argument:
>
>> I want to take my son to nursery school. What's the difference between what I have and what I want? One of distance. What changes distance? My automobile. My automobile won't work. What is needed to make it work? A new battery. What has new batteries? An auto repair shop. I want the repair shop to put in a new battery; but the shop doesn't know I need one. What is the difficulty? One of communication. What allows communication? A telephone . . . and so on.
>
> This kind of analysis—classifying things in terms of the functions they serve and oscillating among ends, functions required, and means that perform them—forms the basic system of heuristic of GPS.

Means–ends analysis is useful, but does not say what to do when several actions will achieve the goal, or when no action will completely achieve it. Arnauld, a follower of Descartes, correctly described a quantitative formula for deciding what action to take in cases like this (see Chapter 16). John Stuart Mill's (1806–1873) book *Utilitarianism* (Mill, 1863) amplifies on this idea. The more formal theory of decisions is discussed in the following section.

[7] In this picture, all meaningful statements can be verified or falsified either by analyzing the meaning of the words or by carrying out experiments. Because this rules out most of metaphysics, as was the intention, logical positivism was unpopular in some circles.

Mathematics (c. 800–present)

Philosophers staked out most of the important ideas of AI, but to make the leap to a formal science required a level of mathematical formalization in three main areas: computation, logic, and probability. The notion of expressing a computation as a formal **algorithm** goes back to al-Khowarazmi, an Arab mathematician of the ninth century, whose writings also introduced Europe to Arabic numerals and algebra.

ALGORITHM

Logic goes back at least to Aristotle, but it was a philosophical rather than mathematical subject until George Boole (1815–1864) introduced his formal language for making logical inference in 1847. Boole's approach was incomplete, but good enough that others filled in the gaps. In 1879, Gottlob Frege (1848–1925) produced a logic that, except for some notational changes, forms the first-order logic that is used today as the most basic knowledge representation system.[8] Alfred Tarski (1902–1983) introduced a theory of reference that shows how to relate the objects in a logic to objects in the real world. The next step was to determine the limits of what could be done with logic and computation.

David Hilbert (1862–1943), a great mathematician in his own right, is most remembered for the problems he did not solve. In 1900, he presented a list of 23 problems that he correctly predicted would occupy mathematicians for the bulk of the century. The final problem asks if there is an algorithm for deciding the truth of any logical proposition involving the natural numbers—the famous *Entscheidungsproblem*, or decision problem. Essentially, Hilbert was asking if there were fundamental limits to the power of effective proof procedures. In 1930, Kurt Gödel (1906–1978) showed that there exists an effective procedure to prove any true statement in the first-order logic of Frege and Russell; but first-order logic could not capture the principle of mathematical induction needed to characterize the natural numbers. In 1931, he showed that real limits do exist. His **incompleteness theorem** showed that in any language expressive enough to describe the properties of the natural numbers, there are true statements that are undecidable: their truth cannot be established by any algorithm.

INCOMPLETENESS THEOREM

This fundamental result can also be interpreted as showing that there are some functions on the integers that cannot be represented by an algorithm—that is, they cannot be computed. This motivated Alan Turing (1912–1954) to try to characterize exactly which functions *are* capable of being computed. This notion is actually slightly problematic, because the notion of a computation or effective procedure really cannot be given a formal definition. However, the Church–Turing thesis, which states that the Turing machine (Turing, 1936) is capable of computing any computable function, is generally accepted as providing a sufficient definition. Turing also showed that there were some functions that no Turing machine can compute. For example, no machine can tell *in general* whether a given program will return an answer on a given input, or run forever.

Although undecidability and noncomputability are important to an understanding of computation, the notion of **intractability** has had a much greater impact. Roughly speaking, a class of problems is called intractable if the time required to solve instances of the class grows at least exponentially with the size of the instances. The distinction between polynomial and exponential growth in complexity was first emphasized in the mid-1960s (Cobham, 1964; Edmonds, 1965). It is important because exponential growth means that even moderate-sized in-

INTRACTABILITY

8 To understand why Frege's notation was not universally adopted, see the cover of this book.

stances cannot be solved in any reasonable time. Therefore, one should strive to divide the overall problem of generating intelligent behavior into tractable subproblems rather than intractable ones.

REDUCTION

The second important concept in the theory of complexity is **reduction**, which also emerged in the 1960s (Dantzig, 1960; Edmonds, 1962). A reduction is a general transformation from one class of problems to another, such that solutions to the first class can be found by reducing them to problems of the second class and solving the latter problems.

NP-COMPLETENESS

How can one recognize an intractable problem? The theory of **NP-completeness**, pioneered by Steven Cook (1971) and Richard Karp (1972), provides a method. Cook and Karp showed the existence of large classes of canonical combinatorial search and reasoning problems that are NP-complete. Any problem class to which an NP-complete problem class can be reduced is likely to be intractable. (Although it has not yet been proved that NP-complete problems are necessarily intractable, few theoreticians believe otherwise.) These results contrast sharply with the "Electronic Super-Brain" enthusiasm accompanying the advent of computers. Despite the ever-increasing speed of computers, subtlety and careful use of resources will characterize intelligent systems. Put crudely, the world is an *extremely* large problem instance!

Besides logic and computation, the third great contribution of mathematics to AI is the theory of probability. The Italian Gerolamo Cardano (1501–1576) first framed the idea of probability, describing it in terms of the possible outcomes of gambling events. Before his time, the outcomes of gambling games were seen as the will of the gods rather than the whim of chance. Probability quickly became an invaluable part of all the quantitative sciences, helping to deal with uncertain measurements and incomplete theories. Pierre Fermat (1601–1665), Blaise Pascal (1623–1662), James Bernoulli (1654–1705), Pierre Laplace (1749–1827), and others advanced the theory and introduced new statistical methods. Bernoulli also framed an alternative view of probability, as a subjective "degree of belief" rather than an objective ratio of outcomes. Subjective probabilities therefore can be updated as new evidence is obtained. Thomas Bayes (1702–1761) proposed a rule for updating subjective probabilities in the light of new evidence (published posthumously in 1763). Bayes' rule, and the subsequent field of Bayesian analysis, form the basis of the modern approach to uncertain reasoning in AI systems. Debate still rages between supporters of the objective and subjective views of probability, but it is not clear if the difference has great significance for AI. Both versions obey the same set of axioms. Savage's (1954) *Foundations of Statistics* gives a good introduction to the field.

DECISION THEORY

As with logic, a connection must be made between probabilistic reasoning and action. **Decision theory**, pioneered by John Von Neumann and Oskar Morgenstern (1944), combines probability theory with utility theory (which provides a formal and complete framework for specifying the preferences of an agent) to give the first general theory that can distinguish good actions from bad ones. Decision theory is the mathematical successor to utilitarianism, and provides the theoretical basis for many of the agent designs in this book.

Psychology (1879–present)

Scientific psychology can be said to have begun with the work of the German physicist Hermann von Helmholtz (1821–1894) and his student Wilhelm Wundt (1832–1920). Helmholtz applied the scientific method to the study of human vision, and his *Handbook of Physiological Optics* is even now described as "the single most important treatise on the physics and physiology of human vision to this day" (Nalwa, 1993, p.15). In 1879, the same year that Frege launched first-order logic, Wundt opened the first laboratory of experimental psychology at the University of Leipzig. Wundt insisted on carefully controlled experiments in which his workers would perform a perceptual or associative task while introspecting on their thought processes. The careful controls went a long way to make psychology a science, but as the methodology spread, a curious phenomenon arose: each laboratory would report introspective data that just happened to match the theories that were popular in that laboratory. The **behaviorism** movement of John Watson (1878–1958) and Edward Lee Thorndike (1874–1949) rebelled against this subjectivism, rejecting *any* theory involving mental processes on the grounds that introspection could not provide reliable evidence. Behaviorists insisted on studying only objective measures of the percepts (or *stimulus*) given to an animal and its resulting actions (or *response*). Mental constructs such as knowledge, beliefs, goals, and reasoning steps were dismissed as unscientific "folk psychology." Behaviorism discovered a lot about rats and pigeons, but had less success understanding humans. Nevertheless, it had a strong hold on psychology (especially in the United States) from about 1920 to 1960.

BEHAVIORISM

The view that the brain possesses and processes information, which is the principal characteristic of **cognitive psychology**, can be traced back at least to the works of William James[9] (1842–1910). Helmholtz also insisted that perception involved a form of unconscious logical inference. The cognitive viewpoint was largely eclipsed by behaviorism until 1943, when Kenneth Craik published *The Nature of Explanation*. Craik put back the missing mental step between stimulus and response. He claimed that beliefs, goals, and reasoning steps could be useful valid components of a theory of human behavior, and are just as scientific as, say, using pressure and temperature to talk about gases, despite their being made of molecules that have neither. Craik specified the three key steps of a knowledge-based agent: (1) the stimulus must be translated into an internal representation, (2) the representation is manipulated by cognitive processes to derive new internal representations, and (3) these are in turn retranslated back into action. He clearly explained why this was a good design for an agent:

COGNITIVE PSYCHOLOGY

> If the organism carries a "small-scale model" of external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it. (Craik, 1943)

An agent designed this way can, for example, plan a long trip by considering various possible routes, comparing them, and choosing the best one, all before starting the journey. Since the 1960s, the information-processing view has dominated psychology. It it now almost taken for granted among many psychologists that "a cognitive theory should be like a computer program" (Anderson, 1980). By this it is meant that the theory should describe cognition as consisting of well-defined transformation processes operating at the level of the information carried by the input signals.

For most of the early history of AI and cognitive science, no significant distinction was drawn between the two fields, and it was common to see AI programs described as psychological

[9] William James was the brother of novelist Henry James. It is said that Henry wrote fiction as if it were psychology and William wrote psychology as if it were fiction.

results without any claim as to the exact human behavior they were modelling. In the last decade or so, however, the methodological distinctions have become clearer, and most work now falls into one field or the other.

Computer engineering (1940–present)

For artificial intelligence to succeed, we need two things: intelligence and an artifact. The computer has been unanimously acclaimed as the artifact with the best chance of demonstrating intelligence. The modern digital electronic computer was invented independently and almost simultaneously by scientists in three countries embattled in World War II. The first operational modern computer was the Heath Robinson,[10] built in 1940 by Alan Turing's team for the single purpose of deciphering German messages. When the Germans switched to a more sophisticated code, the electromechanical relays in the Robinson proved to be too slow, and a new machine called the Colossus was built from vacuum tubes. It was completed in 1943, and by the end of the war, ten Colossus machines were in everyday use.

The first operational *programmable* computer was the Z-3, the invention of Konrad Zuse in Germany in 1941. Zuse invented floating-point numbers for the Z-3, and went on in 1945 to develop Plankalkul, the first high-level programming language. Although Zuse received some support from the Third Reich to apply his machine to aircraft design, the military hierarchy did not attach as much importance to computing as did its counterpart in Britain.

In the United States, the first *electronic* computer, the ABC, was assembled by John Atanasoff and his graduate student Clifford Berry between 1940 and 1942 at Iowa State University. The project received little support and was abandoned after Atanasoff became involved in military research in Washington. Two other computer projects were started as secret military research: the Mark I, II, and III computers were developed at Harvard by a team under Howard Aiken; and the ENIAC was developed at the University of Pennsylvania by a team including John Mauchly and John Eckert. ENIAC was the first general-purpose, electronic, digital computer. One of its first applications was computing artillery firing tables. A successor, the EDVAC, followed John Von Neumann's suggestion to use a stored program, so that technicians would not have to scurry about changing patch cords to run a new program.

But perhaps the most critical breakthrough was the IBM 701, built in 1952 by Nathaniel Rochester and his group. This was the first computer to yield a profit for its manufacturer. IBM went on to become one of the world's largest corporations, and sales of computers have grown to $150 billion/year. In the United States, the computer industry (including software and services) now accounts for about 10% of the gross national product.

Each generation of computer hardware has brought an increase in speed and capacity, and a decrease in price. Computer engineering has been remarkably successful, regularly doubling performance every two years, with no immediate end in sight for this rate of increase. Massively parallel machines promise to add several more zeros to the overall throughput achievable.

Of course, there were calculating devices before the electronic computer. The abacus is roughly 7000 years old. In the mid-17th century, Blaise Pascal built a mechanical adding

[10] Heath Robinson was a cartoonist famous for his depictions of whimsical and absurdly complicated contraptions for everyday tasks such as buttering toast.

and subtracting machine called the Pascaline. Leibniz improved on this in 1694, building a mechanical device that multiplied by doing repeated addition. Progress stalled for over a century until Charles Babbage (1792–1871) dreamed that logarithm tables could be computed by machine. He designed a machine for this task, but never completed the project. Instead, he turned to the design of the Analytical Engine, for which Babbage invented the ideas of addressable memory, stored programs, and conditional jumps. Although the idea of programmable machines was not new—in 1805, Joseph Marie Jacquard invented a loom that could be programmed using punched cards—Babbage's machine was the first artifact possessing the characteristics necessary for universal computation. Babbage's colleague Ada Lovelace, daughter of the poet Lord Byron, wrote programs for the Analytical Engine and even speculated that the machine could play chess or compose music. Lovelace was the world's first programmer, and the first of many to endure massive cost overruns and to have an ambitious project ultimately abandoned.[11] Babbage's basic design was proven viable by Doron Swade and his colleagues, who built a working model using only the mechanical techniques available at Babbage's time (Swade, 1993). Babbage had the right idea, but lacked the organizational skills to get his machine built.

AI also owes a debt to the software side of computer science, which has supplied the operating systems, programming languages, and tools needed to write modern programs (and papers about them). But this is one area where the debt has been repaid: work in AI has pioneered many ideas that have made their way back to "mainstream" computer science, including time sharing, interactive interpreters, the linked list data type, automatic storage management, and some of the key concepts of object-oriented programming and integrated program development environments with graphical user interfaces.

Linguistics (1957–present)

In 1957, B. F. Skinner published *Verbal Behavior*. This was a comprehensive, detailed account of the behaviorist approach to language learning, written by the foremost expert in the field. But curiously, a review of the book became as well-known as the book itself, and served to almost kill off interest in behaviorism. The author of the review was Noam Chomsky, who had just published a book on his own theory, *Syntactic Structures*. Chomsky showed how the behaviorist theory did not address the notion of creativity in language—it did not explain how a child could understand and make up sentences that he or she had never heard before. Chomsky's theory—based on syntactic models going back to the Indian linguist Panini (c. 350 B.C.)—could explain this, and unlike previous theories, it was formal enough that it could in principle be programmed.

Later developments in linguistics showed the problem to be considerably more complex than it seemed in 1957. Language is ambiguous and leaves much unsaid. This means that understanding language requires an understanding of the subject matter and context, not just an understanding of the structure of sentences. This may seem obvious, but it was not appreciated until the early 1960s. Much of the early work in **knowledge representation** (the study of how to put knowledge into a form that a computer can reason with) was tied to language and informed by research in linguistics, which was connected in turn to decades of work on the philosophical analysis of language.

[11] She also gave her name to Ada, the U.S. Department of Defense's all-purpose programming language.

Modern linguistics and AI were "born" at about the same time, so linguistics does not play a large foundational role in the growth of AI. Instead, the two grew up together, intersecting in a hybrid field called **computational linguistics** or **natural language processing**, which concentrates on the problem of language use.

1.3 THE HISTORY OF ARTIFICIAL INTELLIGENCE

With the background material behind us, we are now ready to outline the development of AI proper. We could do this by identifying loosely defined and overlapping phases in its development, or by chronicling the various different and intertwined conceptual threads that make up the field. In this section, we will take the former approach, at the risk of doing some degree of violence to the real relationships among subfields. The history of each subfield is covered in individual chapters later in the book.

The gestation of artificial intelligence (1943–1956)

The first work that is now generally recognized as AI was done by Warren McCulloch and Walter Pitts (1943). They drew on three sources: knowledge of the basic physiology and function of neurons in the brain; the formal analysis of propositional logic due to Russell and Whitehead; and Turing's theory of computation. They proposed a model of artificial neurons in which each neuron is characterized as being "on" or "off," with a switch to "on" occurring in response to stimulation by a sufficient number of neighboring neurons. The state of a neuron was conceived of as "factually equivalent to a proposition which proposed its adequate stimulus." They showed, for example, that any computable function could be computed by some network of connected neurons, and that all the logical connectives could be implemented by simple net structures. McCulloch and Pitts also suggested that suitably defined networks could learn. Donald Hebb (1949) demonstrated a simple updating rule for modifying the connection strengths between neurons, such that learning could take place.

The work of McCulloch and Pitts was arguably the forerunner of both the logicist tradition in AI and the connectionist tradition. In the early 1950s, Claude Shannon (1950) and Alan Turing (1953) were writing chess programs for von Neumann-style conventional computers.[12] At the same time, two graduate students in the Princeton mathematics department, Marvin Minsky and Dean Edmonds, built the first neural network computer in 1951. The SNARC, as it was called, used 3000 vacuum tubes and a surplus automatic pilot mechanism from a B-24 bomber to simulate a network of 40 neurons. Minsky's Ph.D. committee was skeptical whether this kind of work should be considered mathematics, but von Neumann was on the committee and reportedly said, "If it isn't now it will be someday." Ironically, Minsky was later to prove theorems that contributed to the demise of much of neural network research during the 1970s.

[12] Shannon actually had no real computer to work with, and Turing was eventually denied access to his own team's computers by the British government, on the grounds that research into artificial intelligence was surely frivolous.

Princeton was home to another influential figure in AI, John McCarthy. After graduation, McCarthy moved to Dartmouth College, which was to become the official birthplace of the field. McCarthy convinced Minsky, Claude Shannon, and Nathaniel Rochester to help him bring together U.S. researchers interested in automata theory, neural nets, and the study of intelligence. They organized a two-month workshop at Dartmouth in the summer of 1956. All together there were ten attendees, including Trenchard More from Princeton, Arthur Samuel from IBM, and Ray Solomonoff and Oliver Selfridge from MIT.

Two researchers from Carnegie Tech,[13] Allen Newell and Herbert Simon, rather stole the show. Although the others had ideas and in some cases programs for particular applications such as checkers, Newell and Simon already had a reasoning program, the Logic Theorist (LT), about which Simon claimed, "We have invented a computer program capable of thinking non-numerically, and thereby solved the venerable mind–body problem."[14] Soon after the workshop, the program was able to prove most of the theorems in Chapter 2 of Russell and Whitehead's *Principia Mathematica*. Russell was reportedly delighted when Simon showed him that the program had come up with a proof for one theorem that was shorter than the one in *Principia*. The editors of the *Journal of Symbolic Logic* were less impressed; they rejected a paper coauthored by Newell, Simon, and Logic Theorist.

The Dartmouth workshop did not lead to any new breakthroughs, but it did introduce all the major figures to each other. For the next 20 years, the field would be dominated by these people and their students and colleagues at MIT, CMU, Stanford, and IBM. Perhaps the most lasting thing to come out of the workshop was an agreement to adopt McCarthy's new name for the field: **artificial intelligence**.

Early enthusiasm, great expectations (1952–1969)

The early years of AI were full of successes—in a limited way. Given the primitive computers and programming tools of the time, and the fact that only a few years earlier computers were seen as things that could do arithmetic and no more, it was astonishing whenever a computer did anything remotely clever. The intellectual establishment, by and large, preferred to believe that "a machine can never do X" (see Chapter 26 for a long list of X's gathered by Turing). AI researchers naturally responded by demonstrating one X after another. Some modern AI researchers refer to this period as the "Look, Ma, no hands!" era.

Newell and Simon's early success was followed up with the General Problem Solver, or GPS. Unlike Logic Theorist, this program was designed from the start to imitate human problem-solving protocols. Within the limited class of puzzles it could handle, it turned out that the order in which the program considered subgoals and possible actions was similar to the way humans approached the same problems. Thus, GPS was probably the first program to embody the "thinking humanly" approach. The combination of AI and cognitive science has continued at CMU up to the present day.

[13] Now Carnegie Mellon University (CMU).

[14] Newell and Simon also invented a list-processing language, IPL, to write LT. They had no compiler, and translated it into machine code by hand. To avoid errors, they worked in parallel, calling out binary numbers to each other as they wrote each instruction to make sure they agreed.

At IBM, Nathaniel Rochester and his colleagues produced some of the first AI programs. Herbert Gelernter (1959) constructed the Geometry Theorem Prover. Like the Logic Theorist, it proved theorems using explicitly represented axioms. Gelernter soon found that there were too many possible reasoning paths to follow, most of which turned out to be dead ends. To help focus the search, he added the capability to create a numerical representation of a diagram—a particular case of the general theorem to be proved. Before the program tried to prove something, it could first check the diagram to see if it was true in the particular case.

Starting in 1952, Arthur Samuel wrote a series of programs for checkers (draughts) that eventually learned to play tournament-level checkers. Along the way, he disproved the idea that computers can only do what they are told to, as his program quickly learned to play a better game than its creator. The program was demonstrated on television in February 1956, creating a very strong impression. Like Turing, Samuel had trouble finding computer time. Working at night, he used machines that were still on the testing floor at IBM's manufacturing plant. Chapter 5 covers game playing, and Chapter 20 describes and expands on the learning techniques used by Samuel.

John McCarthy moved from Dartmouth to MIT and there made three crucial contributions in one historic year: 1958. In MIT AI Lab Memo No. 1, McCarthy defined the high-level language

LISP

Lisp, which was to become the dominant AI programming language. Lisp is the second-oldest language in current use.[15] With Lisp, McCarthy had the tool he needed, but access to scarce and expensive computing resources was also a serious problem. Thus, he and others at MIT invented time sharing. After getting an experimental time-sharing system up at MIT, McCarthy eventually attracted the interest of a group of MIT grads who formed Digital Equipment Corporation, which was to become the world's second largest computer manufacturer, thanks to their time-sharing minicomputers. Also in 1958, McCarthy published a paper entitled *Programs with Common Sense*, in which he described the Advice Taker, a hypothetical program that can be seen as the first complete AI system. Like the Logic Theorist and Geometry Theorem Prover, McCarthy's program was designed to use knowledge to search for solutions to problems. But unlike the others, it was to embody general knowledge of the world. For example, he showed how some simple axioms would enable the program to generate a plan to drive to the airport to catch a plane. The program was also designed so that it could accept new axioms in the normal course of operation, thereby allowing it to achieve competence in new areas *without being reprogrammed*. The Advice Taker thus embodied the central principles of knowledge representation and reasoning: that it is useful to have a formal, explicit representation of the world and the way an agent's actions affect the world, and to be able to manipulate these representations with deductive processes. It is remarkable how much of the 1958 paper remains relevant after more than 35 years.

1958 also marked the year that Marvin Minsky moved to MIT. For years he and McCarthy were inseparable as they defined the field together. But they grew apart as McCarthy stressed representation and reasoning in formal logic, whereas Minsky was more interested in getting programs to work, and eventually developed an anti-logical outlook. In 1963, McCarthy took the opportunity to go to Stanford and start the AI lab there. His research agenda of using logic to build the ultimate Advice Taker was advanced by J. A. Robinson's discovery of the resolution method (a complete theorem-proving algorithm for first-order logic; see Section 9.6). Work at Stanford emphasized general-purpose methods for logical reasoning. Applications of

[15] FORTRAN is one year older than Lisp.

logic included Cordell Green's question answering and planning systems (Green, 1969b), and the Shakey robotics project at the new Stanford Research Institute (SRI). The latter project, discussed further in Chapter 25, was the first to demonstrate the complete integration of logical reasoning and physical activity.

MICROWORLDS

Minsky supervised a series of students who chose limited problems that appeared to require intelligence to solve. These limited domains became known as **microworlds**. James Slagle's SAINT program (1963a) was able to solve closed-form integration problems typical of first-year college calculus courses. Tom Evans's ANALOGY program (1968) solved geometric analogy problems that appear in IQ tests, such as the one in Figure 1.2. Bertram Raphael's (1968) SIR (Semantic Information Retrieval) was able to accept input statements in a very restricted subset of English and answer questions thereon. Daniel Bobrow's STUDENT program (1967) solved algebra story problems such as

> If the number of customers Tom gets is twice the square of 20 percent of the number of advertisements he runs, and the number of advertisements he runs is 45, what is the number of customers Tom gets?

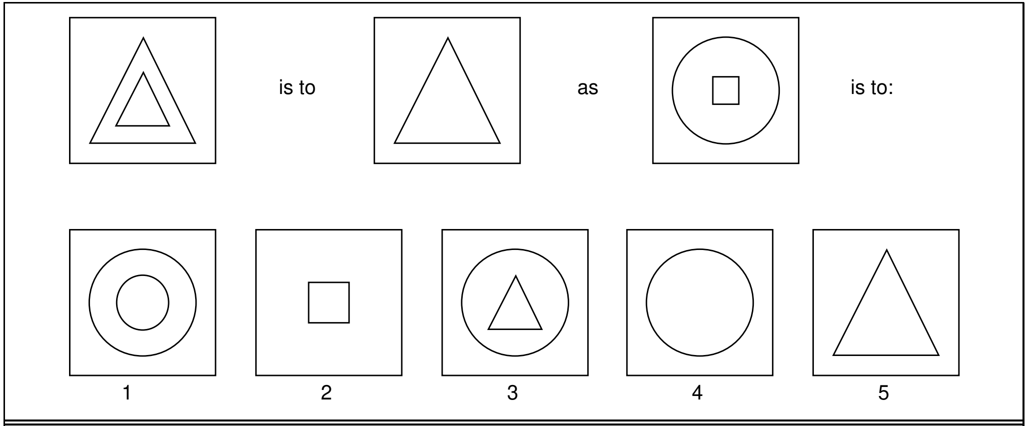

Figure 1.2 An example problem solved by Evans's ANALOGY program.

The most famous microworld was the blocks world, which consists of a set of solid blocks placed on a tabletop (or more often, a simulation of a tabletop), as shown in Figure 1.3. A task in this world is to rearrange the blocks in a certain way, using a robot hand that can pick up one block at a time. The blocks world was home to the vision project of David Huffman (1971), the vision and constraint-propagation work of David Waltz (1975), the learning theory of Patrick Winston (1970), the natural language understanding program of Terry Winograd (1972), and the planner of Scott Fahlman (1974).

Early work building on the neural networks of McCulloch and Pitts also flourished. The work of Winograd and Cowan (1963) showed how a large number of elements could collectively represent an individual concept, with a corresponding increase in robustness and parallelism. Hebb's learning methods were enhanced by Bernie Widrow (Widrow and Hoff, 1960; Widrow, 1962), who called his networks **adalines**, and by Frank Rosenblatt (1962) with his **perceptrons**.

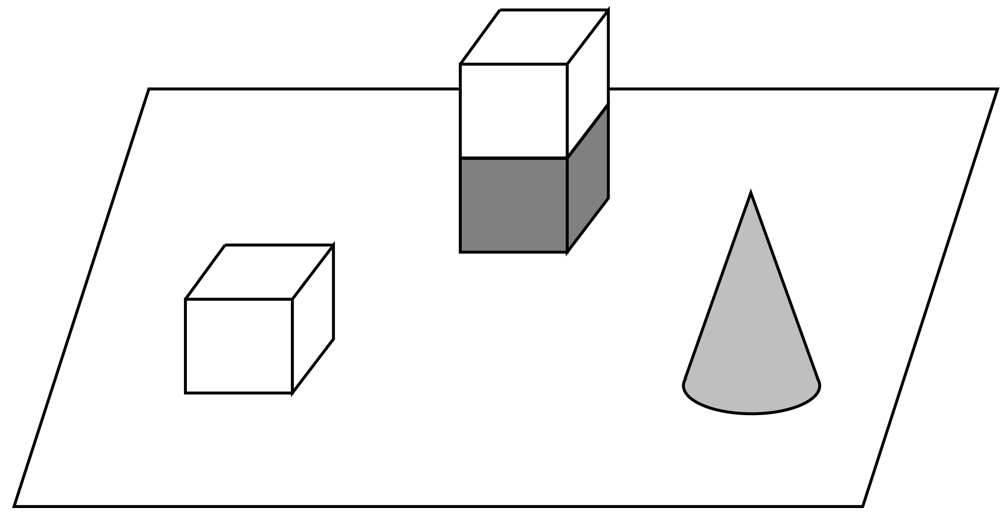

Figure 1.3 A scene from the blocks world. A task for the robot might be "Pick up a big red block," expressed either in natural language or in a formal notation.

Rosenblatt proved the famous **perceptron convergence theorem**, showing that his learning algorithm could adjust the connection strengths of a perceptron to match any input data, provided such a match existed. These topics are covered in Section 19.3.

A dose of reality (1966–1974)

From the beginning, AI researchers were not shy in making predictions of their coming successes. The following statement by Herbert Simon in 1957 is often quoted:

> It is not my aim to surprise or shock you—but the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until—in a visible future—the range of problems they can handle will be coextensive with the range to which human mind has been applied.

Although one might argue that terms such as "visible future" can be interpreted in various ways, some of Simon's predictions were more concrete. In 1958, he predicted that within 10 years a computer would be chess champion, and an important new mathematical theorem would be proved by machine. Claims such as these turned out to be wildly optimistic. The barrier that faced almost all AI research projects was that methods that sufficed for demonstrations on one or two simple examples turned out to fail miserably when tried out on wider selections of problems and on more difficult problems.

The first kind of difficulty arose because early programs often contained little or no knowledge of their subject matter, and succeeded by means of simple syntactic manipulations. Weizenbaum's ELIZA program (1965), which could apparently engage in serious conversation on any topic, actually just borrowed and manipulated the sentences typed into it by a human. A typical story occurred in early machine translation efforts, which were generously funded by the National Research Council in an attempt to speed up the translation of Russian scientific papers in the wake of the Sputnik launch in 1957. It was thought initially that simple syntactic transformations based on the grammars of Russian and English, and word replacement using an electronic dictionary, would suffice to preserve the exact meanings of sentences. In fact, translation requires general knowledge of the subject matter in order to resolve ambiguity and establish the content of the sentence. The famous retranslation of "the spirit is willing but the flesh is weak" as "the vodka is good but the meat is rotten" illustrates the difficulties encountered. In 1966, a report by an advisory committee found that "there has been no machine translation of general scientific text, and none is in immediate prospect." All U.S. government funding for academic translation projects was cancelled.

The second kind of difficulty was the intractability of many of the problems that AI was attempting to solve. Most of the early AI programs worked by representing the basic facts about a problem and trying out a series of steps to solve it, combining different combinations of steps until the right one was found. The early programs were feasible only because microworlds contained very few objects. Before the theory of NP-completeness was developed, it was widely thought that "scaling up" to larger problems was simply a matter of faster hardware and larger memories. The optimism that accompanied the development of resolution theorem proving, for example, was soon dampened when researchers failed to prove theorems involving more than a few dozen facts. *The fact that a program can find a solution in principle does not mean that the program contains any of the mechanisms needed to find it in practice.*

MACHINE EVOLUTION

The illusion of unlimited computational power was not confined to problem-solving programs. Early experiments in **machine evolution** (now called **genetic algorithms**) (Friedberg, 1958; Friedberg *et al.*, 1959) were based on the undoubtedly correct belief that by making an appropriate series of small mutations to a machine code program, one can generate a program with good performance for any particular simple task. The idea, then, was to try random mutations and then apply a selection process to preserve mutations that seemed to improve behavior. Despite thousands of hours of CPU time, almost no progress was demonstrated.

Failure to come to grips with the "combinatorial explosion" was one of the main criticisms of AI contained in the Lighthill report (Lighthill, 1973), which formed the basis for the decision by the British government to end support for AI research in all but two universities. (Oral tradition paints a somewhat different and more colorful picture, with political ambitions and personal animosities that cannot be put in print.)

A third difficulty arose because of some fundamental limitations on the basic structures being used to generate intelligent behavior. For example, in 1969, Minsky and Papert's book *Perceptrons* (1969) proved that although perceptrons could be shown to learn anything they were capable of representing, they could represent very little. In particular, a two-input perceptron could not be trained to recognize when its two inputs were different. Although their results did not apply to more complex, multilayer networks, research funding for neural net research soon dwindled to almost nothing. Ironically, the new back-propagation learning algorithms for multilayer networks that were to cause an enormous resurgence in neural net research in the late 1980s were actually discovered first in 1969 (Bryson and Ho, 1969).

Knowledge-based systems: The key to power? (1969–1979)

The picture of problem solving that had arisen during the first decade of AI research was of a general-purpose search mechanism trying to string together elementary reasoning steps to find complete solutions. Such approaches have been called **weak methods**, because they use weak information about the domain. For many complex domains, it turns out that their performance is also weak. The only way around this is to use knowledge more suited to making larger reasoning steps and to solving typically occurring cases in narrow areas of expertise. One might say that to solve a hard problem, you almost have to know the answer already.

WEAK METHODS

The DENDRAL program (Buchanan *et al.*, 1969) was an early example of this approach. It was developed at Stanford, where Ed Feigenbaum (a former student of Herbert Simon), Bruce Buchanan (a philosopher turned computer scientist), and Joshua Lederberg (a Nobel laureate geneticist) teamed up to solve the problem of inferring molecular structure from the information provided by a mass spectrometer. The input to the program consists of the elementary formula of the molecule (e.g., $C_6H_{13}NO_2$), and the mass spectrum giving the masses of the various fragments of the molecule generated when it is bombarded by an electron beam. For example, the mass spectrum might contain a peak at $m = 15$ corresponding to the mass of a methyl (CH_3) fragment.

The naive version of the program generated all possible structures consistent with the formula, and then predicted what mass spectrum would be observed for each, comparing this with the actual spectrum. As one might expect, this rapidly became intractable for decent-sized molecules. The DENDRAL researchers consulted analytical chemists and found that they worked by looking for well-known patterns of peaks in the spectrum that suggested common substructures in the molecule. For example, the following rule is used to recognize a ketone (C=O) subgroup:

if there are two peaks at x_1 and x_2 such that
(a) $x_1 + x_2 = M + 28$ (M is the mass of the whole molecule);
(b) $x_1 - 28$ is a high peak;
(c) $x_2 - 28$ is a high peak;
(d) At least one of x_1 and x_2 is high.
then there is a ketone subgroup

Having recognized that the molecule contains a particular substructure, the number of possible candidates is enormously reduced. The DENDRAL team concluded that the new system was powerful because

> All the relevant theoretical knowledge to solve these problems has been mapped over from its general form in the [spectrum prediction component] ("first principles") to efficient special forms ("cookbook recipes"). (Feigenbaum *et al.*, 1971)

The significance of DENDRAL was that it was arguably the first successful *knowledge-intensive* system: its expertise derived from large numbers of special-purpose rules. Later systems also incorporated the main theme of McCarthy's Advice Taker approach— the clean separation of the knowledge (in the form of rules) and the reasoning component.

With this lesson in mind, Feigenbaum and others at Stanford began the Heuristic Programming Project (HPP), to investigate the extent to which the new methodology of **expert systems** could be applied to other areas of human expertise. The next major effort was in the area of medical diagnosis. Feigenbaum, Buchanan, and Dr. Edward Shortliffe developed MYCIN to diagnose blood infections. With about 450 rules, MYCIN was able to perform as well as some

EXPERT SYSTEMS

experts, and considerably better than junior doctors. It also contained two major differences from DENDRAL. First, unlike the DENDRAL rules, no general theoretical model existed from which the MYCIN rules could be deduced. They had to be acquired from extensive interviewing of experts, who in turn acquired them from direct experience of cases. Second, the rules had to reflect the uncertainty associated with medical knowledge. MYCIN incorporated a calculus of uncertainty called **certainty factors** (see Chapter 14), which seemed (at the time) to fit well with how doctors assessed the impact of evidence on the diagnosis.

Other approaches to medical diagnosis were also followed. At Rutgers University, Saul Amarel's *Computers in Biomedicine* project began an ambitious attempt to diagnose diseases based on explicit knowledge of the causal mechanisms of the disease process. Meanwhile, large groups at MIT and the New England Medical Center were pursuing an approach to diagnosis and treatment based on the theories of probability and utility. Their aim was to build systems that gave provably optimal medical recommendations. In medicine, the Stanford approach using rules provided by doctors proved more popular at first. But another probabilistic reasoning system, PROSPECTOR (Duda *et al.*, 1979), generated enormous publicity by recommending exploratory drilling at a geological site that proved to contain a large molybdenum deposit.

The importance of domain knowledge was also apparent in the area of understanding natural language. Although Winograd's SHRDLU system for understanding natural language had engendered a good deal of excitement, its dependence on syntactic analysis caused some of the same problems as occurred in the early machine translation work. It was able to overcome ambiguity and understand pronoun references, but this was mainly because it was designed specifically for one area—the blocks world. Several researchers, including Eugene Charniak, a fellow graduate student of Winograd's at MIT, suggested that robust language understanding would require general knowledge about the world and a general method for using that knowledge.

At Yale, the linguist-turned-AI-researcher Roger Schank emphasized this point by claiming, "There is no such thing as syntax," which upset a lot of linguists, but did serve to start a useful discussion. Schank and his students built a series of programs (Schank and Abelson, 1977; Schank and Riesbeck, 1981; Dyer, 1983) that all had the task of understanding natural language. The emphasis, however, was less on language *per se* and more on the problems of representing and reasoning with the knowledge required for language understanding. The problems included representing stereotypical situations (Cullingford, 1981), describing human memory organization (Rieger, 1976; Kolodner, 1983), and understanding plans and goals (Wilensky, 1983). William Woods (1973) built the LUNAR system, which allowed geologists to ask questions in English about the rock samples brought back by the Apollo moon mission. LUNAR was the first natural language program that was used by people other than the system's author to get real work done. Since then, many natural language programs have been used as interfaces to databases.

The widespread growth of applications to real-world problems caused a concomitant increase in the demands for workable knowledge representation schemes. A large number of different representation languages were developed. Some were based on logic—for example, the Prolog language became popular in Europe, and the PLANNER family in the United States. Others, following Minsky's idea of **frames** (1975), adopted a rather more structured approach, collecting together facts about particular object and event types, and arranging the types into a large taxonomic hierarchy analogous to a biological taxonomy.

FRAMES

AI becomes an industry (1980–1988)

The first successful commercial expert system, R1, began operation at Digital Equipment Corporation (McDermott, 1982). The program helped configure orders for new computer systems, and by 1986, it was saving the company an estimated $40 million a year. By 1988, DEC's AI group had 40 deployed expert systems, with more on the way. Du Pont had 100 in use and 500 in development, saving an estimated $10 million a year. Nearly every major U.S. corporation had its own AI group and was either using or investigating expert system technology.

In 1981, the Japanese announced the "Fifth Generation" project, a 10-year plan to build intelligent computers running Prolog in much the same way that ordinary computers run machine code. The idea was that with the ability to make millions of inferences per second, computers would be able to take advantage of vast stores of rules. The project proposed to achieve full-scale natural language understanding, among other ambitious goals.

The Fifth Generation project fueled interest in AI, and by taking advantage of fears of Japanese domination, researchers and corporations were able to generate support for a similar investment in the United States. The Microelectronics and Computer Technology Corporation (MCC) was formed as a research consortium to counter the Japanese project. In Britain, the Alvey report reinstated the funding that was cut by the Lighthill report.[16] In both cases, AI was part of a broad effort, including chip design and human-interface research.

The booming AI industry also included companies such as Carnegie Group, Inference, Intellicorp, and Teknowledge that offered the software tools to build expert systems, and hardware companies such as Lisp Machines Inc., Texas Instruments, Symbolics, and Xerox that were building workstations optimized for the development of Lisp programs. Over a hundred companies built industrial robotic vision systems. Overall, the industry went from a few million in sales in 1980 to $2 billion in 1988.

The return of neural networks (1986–present)

Although computer science had neglected the field of neural networks after Minsky and Papert's *Perceptrons* book, work had continued in other fields, particularly physics. Large collections of simple neurons could be understood in much the same way as large collections of atoms in solids. Physicists such as Hopfield (1982) used techniques from statistical mechanics to analyze the storage and optimization properties of networks, leading to significant cross-fertilization of ideas. Psychologists including David Rumelhart and Geoff Hinton continued the study of neural net models of memory. As we discuss in Chapter 19, the real impetus came in the mid-1980s when at least four different groups reinvented the back-propagation learning algorithm first found in 1969 by Bryson and Ho. The algorithm was applied to many learning problems in computer science and psychology, and the widespread dissemination of the results in the collection *Parallel Distributed Processing* (Rumelhart and McClelland, 1986) caused great excitement.

At about the same time, some disillusionment was occurring concerning the applicability of the expert system technology derived from MYCIN-type systems. Many corporations and

[16] To save embarrassment, a new field called IKBS (Intelligent Knowledge-Based Systems) was defined because Artificial Intelligence had been officially cancelled.

research groups found that building a successful expert system involved much more than simply buying a reasoning system and filling it with rules. Some predicted an "AI Winter" in which AI funding would be squeezed severely. It was perhaps this fear, and the historical factors on the neural network side, that led to a period in which neural networks and traditional AI were seen as rival fields, rather than as mutually supporting approaches to the same problem.

Recent events (1987–present)

Recent years have seen a sea change in both the content and the methodology of research in artificial intelligence.[17] It is now more common to build on existing theories than to propose brand new ones, to base claims on rigorous theorems or hard experimental evidence rather than on intuition, and to show relevance to real-world applications rather than toy examples.

The field of speech recognition illustrates the pattern. In the 1970s, a wide variety of different architectures and approaches were tried. Many of these were rather *ad hoc* and fragile, and were demonstrated on a few specially selected examples. In recent years, approaches based on **hidden Markov models** (HMMs) have come to dominate the area. Two aspects of HMMs are relevant to the present discussion. First, they are based on a rigorous mathematical theory. This has allowed speech researchers to build on several decades of mathematical results developed in other fields. Second, they are generated by a process of training on a large corpus of real speech data. This ensures that the performance is robust, and in rigorous blind tests the HMMs have been steadily improving their scores. Speech technology and the related field of handwritten character recognition are already making the transition to widespread industrial and consumer applications.

Another area that seems to have benefitted from formalization is planning. Early work by Austin Tate (1977), followed up by David Chapman (1987), has resulted in an elegant synthesis of existing planning programs into a simple framework. There have been a number of advances that built upon each other rather than starting from scratch each time. The result is that planning systems that were only good for microworlds in the 1970s are now used for scheduling of factory work and space missions, among other things. See Chapters 11 and 12 for more details.

Judea Pearl's (1988) *Probabilistic Reasoning in Intelligent Systems* marked a new acceptance of probability and decision theory in AI, following a resurgence of interest epitomized by Peter Cheeseman's (1985) article "In Defense of Probability." The **belief network** formalism was invented to allow efficient reasoning about the combination of uncertain evidence. This approach largely overcomes the problems with probabilistic reasoning systems of the 1960s and 1970s, and has come to dominate AI research on uncertain reasoning and expert systems. Work by Judea Pearl (1982a) and by Eric Horvitz and David Heckerman (Horvitz and Heckerman, 1986; Horvitz *et al.*, 1986) promoted the idea of *normative* expert systems: ones that act rationally according to the laws of decision theory and do not try to imitate human experts. Chapters 14 to 16 cover this area.

[17] Some have characterized this change as a victory of the **neats**—those who think that AI theories should be grounded in mathematical rigor—over the **scruffies**—those who would rather try out lots of ideas, write some programs, and then assess what seems to be working. Both approaches are important. A shift toward increased neatness implies that the field has reached a level of stability and maturity. (Whether that stability will be disrupted by a new scruffy idea is another question.)

Similar gentle revolutions have occurred in robotics, computer vision, machine learning (including neural networks), and knowledge representation. A better understanding of the problems and their complexity properties, combined with increased mathematical sophistication, has led to workable research agendas and robust methods. Perhaps encouraged by the progress in solving the subproblems of AI, researchers have also started to look at the "whole agent" problem again. The work of Allen Newell, John Laird, and Paul Rosenbloom on SOAR (Newell, 1990; Laird *et al.*, 1987) is the best-known example of a complete agent architecture in AI. The so-called "situated" movement aims to understand the workings of agents embedded in real environments with continuous sensory inputs. Many interesting results are coming out of such work, including the realization that the previously isolated subfields of AI may need to be reorganized somewhat when their results are to be tied together into a single agent design.

1.4 THE STATE OF THE ART

International grandmaster Arnold Denker studies the pieces on the board in front of him. He realizes there is no hope; he must resign the game. His opponent, HITECH, becomes the first computer program to defeat a grandmaster in a game of chess (Berliner, 1989).

"I want to go from Boston to San Francisco," the traveller says into the microphone. "What date will you be travelling on?" is the reply. The traveller explains she wants to go October 20th, nonstop, on the cheapest available fare, returning on Sunday. A speech understanding program named PEGASUS handles the whole transaction, which results in a confirmed reservation that saves the traveller $894 over the regular coach fare. Even though the speech recognizer gets one out of ten words wrong,[18] it is able to recover from these errors because of its understanding of how dialogs are put together (Zue *et al.*, 1994).

An analyst in the Mission Operations room of the Jet Propulsion Laboratory suddenly starts paying attention. A red message has flashed onto the screen indicating an "anomaly" with the Voyager spacecraft, which is somewhere in the vicinity of Neptune. Fortunately, the analyst is able to correct the problem from the ground. Operations personnel believe the problem might have been overlooked had it not been for MARVEL, a real-time expert system that monitors the massive stream of data transmitted by the spacecraft, handling routine tasks and alerting the analysts to more serious problems (Schwuttke, 1992).

Cruising the highway outside of Pittsburgh at a comfortable 55 mph, the man in the driver's seat seems relaxed. He should be—for the past 90 miles, he has not had to touch the steering wheel. The real driver is a robotic system that gathers input from video cameras, sonar, and laser range finders attached to the van. It combines these inputs with experience learned from training runs and succesfully computes how to steer the vehicle (Pomerleau, 1993).

A leading expert on lymph-node pathology describes a fiendishly difficult case to the expert system, and examines the system's diagnosis. He scoffs at the system's response. Only slightly worried, the creators of the system suggest he ask the computer for an explanation of

[18] Some other existing systems err only half as often on this task.

the diagnosis. The machine points out the major factors influencing its decision, and explains the subtle interaction of several of the symptoms in this case. The expert admits his error, eventually (Heckerman, 1991).

From a camera perched on a street light above the crossroads, the traffic monitor watches the scene. If any humans were awake to read the main screen, they would see "Citroën 2CV turning from Place de la Concorde into Champs Elysées," "Large truck of unknown make stopped on Place de la Concorde," and so on into the night. And occasionally, "Major incident on Place de la Concorde, speeding van collided with motorcyclist," and an automatic call to the emergency services (King *et al.*, 1993; Koller *et al.*, 1994).

These are just a few examples of artificial intelligence systems that exist today. Not magic or science fiction—but rather science, engineering, and mathematics, to which this book provides an introduction.

1.5 SUMMARY

This chapter defines AI and establishes the cultural background against which it has developed. Some of the important points are as follows:

- Different people think of AI differently. Two important questions to ask are: Are you concerned with thinking or behavior? Do you want to model humans, or work from an ideal standard?
- In this book, we adopt the view that intelligence is concerned mainly with **rational action**. Ideally, an **intelligent agent** takes the best possible action in a situation. We will study the problem of building agents that are intelligent in this sense.
- Philosophers (going back to 400 B.C.) made AI conceivable by considering the ideas that the mind is in some ways like a machine, that it operates on knowledge encoded in some internal language, and that thought can be used to help arrive at the right actions to take.
- Mathematicians provided the tools to manipulate statements of logical certainty as well as uncertain, probabilistic statements. They also set the groundwork for reasoning about algorithms.
- Psychologists strengthened the idea that humans and other animals can be considered information processing machines. Linguists showed that language use fits into this model.
- Computer engineering provided the artifact that makes AI applications possible. AI programs tend to be large, and they could not work without the great advances in speed and memory that the computer industry has provided.
- The history of AI has had cycles of success, misplaced optimism, and resulting cutbacks in enthusiasm and funding. There have also been cycles of introducing new creative approaches and systematically refining the best ones.
- Recent progress in understanding the theoretical basis for intelligence has gone hand in hand with improvements in the capabilities of real systems.

BIBLIOGRAPHICAL AND HISTORICAL NOTES

Daniel Crevier's (1993) *Artificial Intelligence* gives a complete history of the field, and Raymond Kurzweil's (1990) *Age of Intelligent Machines* situates AI in the broader context of computer science and intellectual history in general. Dianne Martin (1993) documents the degree to which early computers were endowed by the media with mythical powers of intelligence.

The methodological status of artificial intelligence is discussed in *The Sciences of the Artificial*, by Herb Simon (1981), which discusses research areas concerned with complex artifacts. It explains how AI can be viewed as both science and mathematics.

Artificial Intelligence: The Very Idea, by John Haugeland (1985) gives a readable account of the philosophical and practical problems of AI. Cognitive science is well-described by Johnson-Laird's *The Computer and the Mind: An Introduction to Cognitive Science*. Baker (1989) covers the syntactic part of modern linguistics, and Chierchia and McConnell-Ginet (1990) cover semantics. Allen (1995) covers linguistics from the AI point of view.

Early AI work is covered in Feigenbaum and Feldman's *Computers and Thought*, Minsky's *Semantic Information Processing*, and the *Machine Intelligence* series edited by Donald Michie. A large number of influential papers are collected in *Readings in Artificial Intelligence* (Webber and Nilsson, 1981). Early papers on neural networks are collected in *Neurocomputing* (Anderson and Rosenfeld, 1988). The *Encyclopedia of AI* (Shapiro, 1992) contains survey articles on almost every topic in AI. These articles usually provide a good entry point into the research literature on each topic. The four-volume *Handbook of Artificial Intelligence* (Barr and Feigenbaum, 1981) contains descriptions of almost every major AI system published before 1981.

The most recent work appears in the proceedings of the major AI conferences: the biennial International Joint Conference on AI (IJCAI), and the annual National Conference on AI, more often known as AAAI, after its sponsoring organization. The major journals for general AI are *Artificial Intelligence*, *Computational Intelligence*, the IEEE *Transactions on Pattern Analysis and Machine Intelligence*, and the electronic *Journal of Artificial Intelligence Research*. There are also many journals devoted to specific areas, which we cover in the appropriate chapters. Commercial products are covered in the magazines *AI Expert* and *PC AI*. The main professional societies for AI are the American Association for Artificial Intelligence (AAAI), the ACM Special Interest Group in Artificial Intelligence (SIGART), and the Society for Artificial Intelligence and Simulation of Behaviour (AISB). AAAI's *AI Magazine* and the *SIGART Bulletin* contain many topical and tutorial articles as well as announcements of conferences and workshops.

EXERCISES

These exercises are intended to stimulate discussion, and some might be set as term projects. Alternatively, preliminary attempts can be made now, and these attempts can be reviewed after completing the book.

1.1 Read Turing's original paper on AI (Turing, 1950). In the paper, he discusses several potential objections to his proposed enterprise and his test for intelligence. Which objections still carry some weight? Are his refutations valid? Can you think of new objections arising from developments since he wrote the paper? In the paper, he predicts that by the year 2000, a computer will have a 30% chance of passing a five-minute Turing Test with an unskilled interrogator. Do you think this is reasonable?

1.2 We characterized the definitions of AI along two dimensions, human vs. ideal and thought vs. action. But there are other dimensions that are worth considering. One dimension is whether we are interested in theoretical results or in practical applications. Another is whether we intend our intelligent computers to be conscious or not. Philosophers have had a lot to say about this issue, and although most AI researchers are happy to leave the questions to the philosophers, there has been heated debate. The claim that machines can be conscious is called the **strong AI** claim; the **weak AI** position makes no such claim. Characterize the eight definitions on page 5 and the seven following definitions according to the four dimensions we have mentioned and whatever other ones you feel are helpful.

STRONG AI

WEAK AI

Artificial intelligence is . . .

a. "a collection of algorithms that are computationally tractable, adequate approximations of intractably specified problems" (Partridge, 1991)

b. "the enterprise of constructing a physical symbol system that can reliably pass the Turing Test" (Ginsberg, 1993)

c. "the field of computer science that studies how machines can be made to act intelligently" (Jackson, 1986)

d. "a field of study that encompasses computational techniques for performing tasks that apparently require intelligence when performed by humans" (Tanimoto, 1990)

e. "a very general investigation of the nature of intelligence and the principles and mechanisms required for understanding or replicating it" (Sharples *et al.*, 1989)

f. "the getting of computers to do things that seem to be intelligent" (Rowe, 1988).

1.3 There are well-known classes of problems that are intractably difficult for computers, and other classes that are provably undecidable by any computer. Does this mean that AI is impossible?

1.4 Suppose we extend Evans's ANALOGY program so that it can score 200 on a standard IQ test. Would we then have a program more intelligent than a human? Explain.

1.5 Examine the AI literature to discover whether or not the following tasks can currently be solved by computers:

a. Playing a decent game of table tennis (ping-pong).

b. Driving in the center of Cairo.

c. Playing a decent game of bridge at a competitive level.

d. Discovering and proving new mathematical theorems.

e. Writing an intentionally funny story.

f. Giving competent legal advice in a specialized area of law.

g. Translating spoken English into spoken Swedish in real time.

For the currently infeasible tasks, try to find out what the difficulties are and estimate when they will be overcome.

1.6 Find an article written by a lay person in a reputable newspaper or magazine claiming the achievement of some intelligent capacity by a machine, where the claim is either wildly exaggerated or false.

1.7 Fact, fiction, and forecast:

- **a**. Find a claim in print by a reputable philosopher or scientist to the effect that a certain capacity will never be exhibited by computers, where that capacity has now been exhibited.
- **b**. Find a claim by a reputable computer scientist to the effect that a certain capacity would be exhibited by a date that has since passed, without the appearance of that capacity.
- **c**. Compare the accuracy of these predictions to predictions in other fields such as biomedicine, fusion power, nanotechnology, transportation, or home electronics.

1.8 Some authors have claimed that perception and motor skills are the most important part of intelligence, and that "higher-level" capacities are necessarily parasitic—simple add-ons to these underlying facilities. Certainly, most of evolution and a large part of the brain have been devoted to perception and motor skills, whereas AI has found tasks such as game playing and logical inference to be easier, in many ways, than perceiving and acting in the real world. Do you think that AI's traditional focus on higher-level cognitive abilities is misplaced?

1.9 "Surely computers cannot be intelligent—they can only do what their programmers tell them." Is the latter statement true, and does it imply the former?

1.10 "Surely animals cannot be intelligent—they can only do what their genes tell them." Is the latter statement true, and does it imply the former?