Computational Intelligence

Michael Jordan and Stuart Russell
Computer Science Division
University of California
Berkeley, CA 94720

June 10, 1998

There are two complementary views of artificial intelligence (AI): as an engineering discipline concerned with the creation of intelligent machines, and as an empirical science concerned with the computational modeling of human intelligence. When the field was young, these two views were seldom distinguished. Since then, a substantial divide has opened up, with the former view dominating modern AI and the latter view characterizing much of modern cognitive science. For this reason, we have adopted the more neutral term "computational intelligence" as the title of this article—both communities are attacking the problem of understanding intelligence in computational terms.

It is our belief that the differences between the engineering models and the cognitively inspired models are, in fact, small compared to the vast gulf in competence between these models and human levels of intelligence. For humans are, to a first approximation, *intelligent*; they can perceive, act, learn, reason, and communicate *successfully* despite the enormous difficulty of these tasks. Indeed, we expect that as further progress is made in trying to emulate this success, the engineering and cognitive models will become more similar. Already, the traditionally antagonistic "connectionist" and "symbolic" camps are finding common ground, particularly in their understanding of reasoning under uncertainty and learning. This sort of cross-fertilization was a central aspect of the early vision of cognitive science as an interdisciplinary enterprise.

Machines and Cognition. The conceptual precursors of AI can be traced back many centuries. LOGIC, the formal theory of deductive reasoning, was studied in ancient Greece, as were ALGORITHMS for mathematical computations. In the late 17th century, Wilhelm Leibniz actually constructed simple "conceptual calculators," but their representational and combinatorial powers

were far too limited. In the 19th century, Charles Babbage designed (but did not build) a device capable of universal computation, and his collaborator Ada Lovelace speculated that the machine might one day be programmed to play chess or compose music. Fundamental work by Alan TURING in the 1930s formalized the notion of universal computation; the famous CHURCH-TURING THESIS proposed that all sufficiently powerful computing devices were essentially identical in the sense that any one device could emulate the operations of any other. From here it was a small step to the bold hypothesis that human cognition was a form of COMPUTATION in exactly this sense, and could therefore be emulated by computers.

By this time, neurophysiology had already established that the brain consisted largely of a vast interconnected network of NEURONS that used some form of electrical signalling mechanism. The first mathematical model relating computation and the brain appeared in a seminal paper entitled "A logical calculus of the ideas immanent in nervous activity," by Warren MCCULLOCH and Walter PITTS (1943). The paper proposed an abstract model of neurons as linear threshold units—logical "gates" that output a signal if the weighted sum of their inputs exceeds a threshold value (see COMPUTING IN SINGLE NEURONS). It was shown that a network of such gates could represent any logical function, and, with suitable delay components to implement memory, would be capable of universal computation. Together with HEBB's model of learning in networks of neurons, this work can be seen as a precursor of modern NEURAL NETWORKS and CONNECTIONIST COGNITIVE MODELING. Its stress on the representation of logical concepts by neurons also provided impetus to the "logicist" view of AI.

The emergence of AI proper as a recognizable field required the availability of usable computers; this resulted from the wartime efforts led by Turing in Britain and by John VON NEUMANN in the United States. It also required a banner to be raised; this was done with relish by Turing's (1950) paper "Computing Machinery and Intelligence," wherein an operational definition for intelligence was proposed (the Turing Test) and many future developments were sketched out.

One should not underestimate the level of controversy surrounding AI's initial phase. The popular press was only too ready to ascribe intelligence to the new "Electronic Super-Brains," but many academics refused to contemplate the idea of intelligent computers. In his 1950 paper, Turing went to great lengths to catalogue and refute many of their objections. Ironically, one objection already voiced by Kurt Gödel, and repeated up to the present day in various forms, rested on the ideas of incompleteness and undecidability in formal systems to which Turing himself had contributed (see GÖDEL's THEOREMS). Other objectors denied the possibility of CONSCIOUSNESS in computers, and with it the possibility of intelligence. Turing explicitly sought to separate the two, focusing

on the objective question of intelligent *behavior* while admitting that consciousness might remain a mystery—as indeed it has.

The next step in the emergence of AI was the formation of a research community; this was achieved at the 1956 Dartmouth meeting convened by John MCCARTHY. Perhaps the most advanced work presented at this meeting was that of Allen NEWELL and Herb SIMON, whose program of research in SYMBOLIC COGNITIVE MODELING was one of the principal influences on cognitive psychology and information-processing psychology. Newell and Simon's IPL languages were the first symbolic programming languages and among the first high-level languages of any kind. McCarthy's LISP language, developed slightly later, soon became the standard programming language of the AI community and in many ways remains unsurpassed even today.

Contemporaneous developments in other fields also led to a dramatic increase in the precision and complexity of the models that could be proposed and analyzed. In LINGUISTICS, for example, work by Chomsky (1957) on formal grammars opened up new avenues for the mathematical modeling of mental structures. Norbert WIENER developed the field of cybernetics (see CONTROL THEORY and MOTOR CONTROL) to provide mathematical tools for the analysis and synthesis of physical control systems. The theory of optimal control in particular has many parallels with the theory of rational agents (see below), but within this tradition no model of internal representation was ever developed.

As might be expected from so young a field with so broad a mandate that draws on so many traditions, the history of AI has been marked by substantial changes in fashion and opinion. Its early days might be described as the "Look, Ma, no hands!" era, when the emphasis was on showing a doubting world that computers *could* play chess, learn, see, and do all the other things thought to be impossible. A wide variety of methods was tried, ranging from general-purpose symbolic problem solvers to simple neural networks. By the late 1960s, a number of practical and theoretical setbacks had convinced most AI researchers that there would be no simple "magic bullet." The general-purpose methods that had initially seemed so promising came to be called *weak methods* because their reliance on extensive combinatorial search and first-principles knowledge could not overcome the complexity barriers that were, by that time, seen as unavoidable. The 1970s saw the rise of an alternative approach based on the application of large amounts of domain-specific knowledge, expressed in forms that were close enough to the explicit solution as to require little additional computation. Ed Feigenbaum's gnomic dictum, "Knowledge is power," was the watchword of the boom in industrial and commercial application of *expert systems* in the early 1980s.

When the first generation of expert system technology turned out to be too fragile for widespread

use, a so-called "AI Winter" set in—government funding of AI and public perception of its promise both withered in the late 1980s. At the same time, a revival of interest in neural network approaches led to the same kind of optimism as had characterized "traditional" AI in the early 1980s. Since that time, very substantial progress has been made in a number of areas within AI, leading to renewed commercial interest in fields such as *data mining* (applied machine learning) and a new wave of expert system technology based on probabilistic inference. The 1990s may in fact come to be seen as the decade of probability. Besides expert systems, the so-called *Bayesian* approach (named after the Rev. Thomas Bayes, 18th-century author of the fundamental rule for probabilistic reasoning) has led to new methods in planning, natural language understanding, and learning. Indeed, it seems likely that work on the latter topic will lead to a reconciliation of symbolic and connectionist views of intelligence.

Artificial Intelligence: What's the Problem? The consensus apparent in modern textbooks (Russell & Norvig, 1995; Poole, Mackworth, & Goebel, 1997; Nilsson, 1998) is that AI is about the design of intelligent *agents*. An agent is an entity that can be understood as perceiving and acting on its environment. An agent is *rational* to the extent that its actions can be expected to achieve its goals, given the information available from its perceptual processes. Whereas the Turing Test defined only an informal notion of intelligence as emulation of humans, the theory of RATIONAL AGENCY provides a first pass at a *formal specification* for intelligent agents, with the possibility of a constructive theory to satisfy this specification. Although the last section of this introduction argues that this specification needs a radical rethinking, the idea of rational decision making has nonetheless been the foundation for most of the current research trends in AI.

The focus on AI as the design of intelligent agents is in fact a fairly recent preoccupation. Until the mid-1980s, most research in "core AI" (that is, AI excluding the areas of robotics and computer vision) concentrated on isolated reasoning tasks whose inputs were provided by humans and whose outputs were interpreted by humans. Mathematical theorem-proving systems, English question-answering systems, and medical expert systems all had this flavor—none of them took actions in any meaningful sense. The so-called "situated" movement in AI (see SITUATEDNESS/EMBEDDEDNESS) stressed the point that reasoning is not an end in itself, but serves the purpose of enabling the selection of actions that will affect the reasoner's environment in desirable ways. Thus, reasoning always occurs in a specific context for specific goals; by removing context and taking responsibility for action selection, AI researchers were in danger of defining a subtask that, although useful, actually had no role in the design of a complete intelligent system.

For example, some early medical expert systems were constructed in such a way as to accept as input a complete list of symptoms and to output the most likely diagnosis. This might seem like a useful tool, but it ignores several key aspects of medicine: the crucial role of hypothesis-directed *gathering* of information, the very complex task of interpreting sensory data to obtain suggestive and uncertain indicators of symptoms, and the overriding goal of *curing* the patient, which may involve treatments aimed at less likely but potentially dangerous conditions rather than more likely but harmless ones. A second example occurred in robotics: much research was done on motion planning under the assumption that the locations and shapes of all objects in the environment were known exactly; yet no feasible vision system can, or should, be designed to obtain this information.

When one thinks about building intelligent agents, it quickly becomes obvious that the task environment in which the agent will operate is a primary determiner of the appropriate design. For example, if all relevant aspects of the environment are immediately available to the agent's perceptual apparatus—as, for example, when playing backgammon—then the environment is said to be *fully observable* and the agent need maintain no internal model of the world at all. Backgammon is also *discrete* as opposed to *continuous*—that is, there is a finite set of distinct backgammon board states, whereas tennis, say, requires real-valued variables and changes continuously over time. Backgammon is is *stochastic* as opposed to *deterministic*, because it includes dice rolls and unpredictable opponents; hence an agent may need to make contingency plans for many possible outcomes. Backgammon, unlike tennis, is also *static* rather than *dynamic*, in that nothing much happens while the agent is deciding what move to make. Finally, the "physical laws" of the backgammon universe—what the legal moves are and what effect they have—are known rather than unknown. These distinctions alone (and there are many more) define 32 substantially different kinds of task environment. This variety of tasks, rather than any true conceptual differences, may be responsible for the variety of computational approaches to intelligence that, on the surface, seem so philosophically incompatible.

Architectures of Cognition. Any computational theory of intelligence must propose, at least implicitly, an INTELLIGENT AGENT ARCHITECTURE. Such an architecture defines the underlying organization of the cognitive processes comprising intelligence, and forms the computational substrate upon which domain-specific capabilities are built. For example, an architecture may provide a generic capability for learning the "physical laws" of the environment, for combining inputs from multiple sensors, or for deliberating about actions by envisioning and evaluating their effects.

There is, as yet, no satisfactory theory that defines the range of possible architectures for intelligent

systems, or identifies the optimal architecture for a given task environment, or provides a reasonable specification of what is required for an architecture to support "general-purpose" intelligence, either in machines or humans. Some researchers see the observed variety of intelligent behaviors as a consequence of the operation of a unified, general-purpose problem-solving architecture (Newell, 1990); others propose a functional division of the architecture, with modules for perception, learning, reasoning, communication, locomotion, and so on (see MODULARITY OF MIND). Evidence from neuroscience (for example, lesion studies) is often interpreted as showing that the brain is divided into areas, each of which performs some function in this sense; yet the functional descriptions (e.g., "language," "face recognition," etc.) are often subjective and informal and the nature of the connections among the components remains obscure. In the absence of deeper theory, such generalizations from scanty evidence must remain highly suspect. That is, the basic organizational principles of intelligence are still up for grabs.

Proposed architectures vary along a number of dimensions. Perhaps the most commonly cited distinction is between "symbolic" and "connectionist" approaches. These approaches are often thought to be based on fundamentally irreconcilable philosophical foundations. We will argue that, to a large extent, they are complementary; where comparable, they form a continuum.

Roughly speaking, a *symbol* is an object, part of the internal state of an agent, that has two properties: it can be compared to other symbols to test for equality, and it can be combined with other symbols to form *symbol structures*. The symbolic approach to AI, in its purest form, is embodied in the Physical Symbol System (PSS) hypothesis (Newell & Simon, 1972), which proposes that algorithmic manipulation of symbol structures is necessary and sufficient for general intelligence. (See also COMPUTATIONAL THEORY OF MIND.)

The PSS hypothesis, if taken to its extreme, is identical to the view that cognition can be understood as COMPUTATION. Symbol systems can emulate any Turing machine; in particular, they can carry out finite-precision numerical operations and thereby implement neural networks. Most AI researchers, however, interpret the PSS hypothesis more narrowly, in the sense that primitive numerical quantities that are manipulated as *magnitudes*, rather than simply tested for (in)equality, are considered to be out of bounds. The Soar architecture (Newell, 1990), which uses PROBLEM SOLVING as its underlying formalism, is the most well-developed instantiation of the pure symbolic approach to cognition (see COGNITIVE MODELING, SYMBOLIC).

The symbolic tradition also encompasses approaches to AI that are based on LOGIC. The symbols in the logical languages are used to represent objects and relations among objects, and symbol structures called *sentences* are used to represent facts that the agent knows. Sentences are manip-

ulated according to certain rules to generate new sentences that follow logically from the original sentences. The details of logical agent design are given below in the section on knowledge-based systems; what is relevant here is the use of symbol structures as direct representations of the world. For example, if the agent sees John sitting on the fence, it might construct an internal representation from symbols that represent John, the fence, and the sitting-on relation. If Mary is on the fence instead, the symbol structure would be the same except for the use of a symbol for Mary instead of John.

This kind of compositionality of representations is characteristic of symbolic approaches. A more restricted kind of compositionality can occur even in much simpler systems. For example, in the network of logical gates proposed by McCulloch and Pitts, we might have a neuron J that is "on" whenever the agent sees John on the fence; and another neuron M that is "on" when Mary is on the fence. Then the proposition "either John or Mary is on the fence" can be represented by a neuron that is connected to J and M with the appropriate connection strengths. We call this kind of representation *propositional*, since the fundamental elements are propositions rather than symbols denoting objects and relations. In the words of McCulloch and Pitts, the state of a neuron was conceived of as "factually equivalent to a proposition which proposed its adequate stimulus." We will also extend the standard sense of "propositional" to cover neural networks comprised of neurons with continuous real-valued activations, rather than the 1/0 activations in the original McCulloch-Pitts threshold neurons.

It is clear that, in this sense, the raw sensory data available to an agent is propositional. For example, the elements of visual perception are "pixels" whose propositional content is, e.g., "this area of my retina is receiving bright red light." This observation leads to the first difficulty for the symbolic approach: how to move from sensory data to symbolic representations. This so-called *symbol grounding problem* has been deemed insoluble by some philosophers (see the article on CONCEPTS), thereby dooming the symbolic approach to oblivion. On the other hand, existence proofs of its solubility abound. For example, Shakey, the first substantial robotics project in AI, used symbolic (logical) reasoning for its deliberations, but interacted with the world quite happily (albeit slowly) through video cameras and wheels (see Raphael, 1976).

A related problem for purely symbolic approaches is that sensory information about the physical world is usually thought of as numerical—light intensities, forces, strains, frequencies, and so on. Thus, there must at least be a layer of nonsymbolic computation between the real world and the realm of pure symbols. Neither the theory nor the practice of symbolic AI argues against the existence of such a layer, but its existence does open up the possibility that some substantial part

of cognition occurs therein without ever reaching the symbolic level.

A deeper problem for the narrow PSS hypothesis is UNCERTAINTY—the unavoidable fact that unreliable and partial sensory information, combined with unreliable and partial theories of how the world works, must leave an agent with some doubt as to the truth of virtually all propositions of interest. For example, the stock market may soon recover this week's losses, or it may not. Whether to buy, sell, or hold depends on one's assessment of the prospects. Similarly, a person spotted across a crowded, smoky night club may or may not be an old friend. Whether to wave in greeting depends on how certain one is (and on one's sensitivity to embarrassment due to waving at complete strangers). Although many decisions under uncertainty can be made without reference to numerical degrees of belief (Wellman, 1990), one has a lingering sense that degrees of belief in propositions may be a fundamental component of our mental representations. Accounts of such phenomena based on probability theory are now widely accepted within AI as an *augmentation* of the purely symbolic view; in particular, probabilistic models are a natural generalization of the logical approach. Recent work has also shown that some connectionist representations (e.g., Boltzmann machines) are essentially identical to probabilistic network models developed in AI (see NEURAL NETWORKS).

The three issues raised in the preceding paragraphs—sensorimotor connections to the external world, handling real-valued inputs and outputs, and robust handling of noisy and uncertain information—are primary motivations for the connectionist approach to cognition. (The existence of networks of neurons in the brain is obviously another.) Neural network models show promise for many low-level tasks such as visual pattern recognition and speech recognition. The most obvious drawback of the connectionist approach is the difficulty of envisaging a means to model higher levels of cognition (see articles on the BINDING PROBLEM and COGNITIVE MODELING, CONNECTIONIST), particularly when compared to the ability of symbol systems to generate an unbounded variety of structures from a finite set of symbols (see COMPOSITIONALITY). Some solutions have been proposed (see, for example, BINDING BY NEURAL SYNCHRONY); these solutions provide a plausible neural *implementation* of symbolic models of cognition, rather than an *alternative*.

Another problem for connectionist and other propositional approaches is the modeling of *temporally extended* behavior. Unless the external environment is completely observable by the agent's sensors, such behavior requires the agent to maintain some internal state information that reflects properties of the external world that are not directly observable. In the symbolic/logical approach, sentences such as "My car is parked at the corner of Columbus and Union" can be stored in "working memory" or in a "temporal knowledge base" and updated as appropriate. In connectionist models, internal

state requires the use of RECURRENT NETWORKS, which are as yet poorly understood.

In summary, the symbolic and connectionist approaches seem not antithetical but complementary—connectionist models may handle low-level cognition and may (or rather *must*, in some form) provide a substrate for higher-level symbolic processes. Probabilistic approaches to representation and reasoning may unify the symbolic and connectionist traditions. It seems that the more relevant distinction is between propositional and more expressive forms of representation.

Related to the symbolic/connectionist debate is the distinction between *deliberative* and *reactive* models of cognition. Most AI researchers view intelligent behavior as resulting, at least in part, from deliberation over possible courses of action based on the agent's knowledge of the world and of the expected results of its actions. This seems self-evident to the average person in the street, but it has always been a controversial hypothesis—in fact, according to BEHAVIORISM, it is meaningless. With the development of KNOWLEDGE-BASED SYSTEMS, starting from the famous "Advice Taker" paper by McCarthy (1958), the deliberative model could be put to the test. The core of a knowledge-based agent is the knowledge base and its associated reasoning procedures; the rest of the design follows straightforwardly. First, we need some way of acquiring the necessary knowledge: this could be from experience through MACHINE LEARNING methods, or from humans and books through NATURAL LANGUAGE PROCESSING, or by direct programming, or through perceptual processes such as MACHINE VISION. Given knowledge of its environment and of its objectives, an agent can reason that certain actions will achieve those objectives and should be executed. At this point, if we are dealing with a physical environment, ROBOTICS takes over, handling the mechanical and geometric aspects of motion and manipulation.

The following sections deal with each of these areas in turn. It should be noted, however, that the story in the preceding paragraph is a gross idealization. It is, in fact, close to the view caricatured as Good Old-Fashioned AI (GOFAI) by John Haugeland (1985) and Hubert Dreyfus (1992). In the five decades since Turing's paper, AI researchers have discovered that attaining real competence is not so simple—the principle barrier being COMPUTATIONAL COMPLEXITY. The idea of *reactive systems* (see also AUTOMATA) is to implement direct mappings from perception to action that avoid the expensive intermediate steps of representation and reasoning. This observation was made within the first month of the Shakey project (Raphael, 1976) and given new life in the field of BEHAVIOR-BASED ROBOTICS (Brooks, 1991). Direct mappings of this kind can be learned from experience or can be compiled from the results of deliberation within a knowledge-based architecture (see EXPLANATION-BASED LEARNING). Most current models propose a *hybrid* agent design incorporating a variety of decision-making mechanisms, perhaps with capabilities for

METAREASONING to control and integrate these mechanisms. Some have even proposed that intelligent systems should be constructed from large numbers of separate agents, each with percepts, actions, and goals of its own (Minsky, 1986)—much as a nation's economy is made up of lots of separate humans. The theory of MULTI-AGENT SYSTEMS explains how, in some cases, the goals of the whole agent can be achieved even when each sub-agent pursues its own ends.

Knowledge-based systems. The *procedural/declarative controversy*, which raged in AI through most of the 1970s, was about which way to build AI systems (see, for example, Boden, 1977). The procedural view held that systems could be constructed by encoding expertise in domain-specific algorithms—for example, a procedure for diagnosing migraines by asking specific sequences of questions. The declarative view, on the other hand, held that systems should be *knowledge-based*, that is, composed from domain-specific *knowledge*—for example, the symptoms typically associated with various ailments—combined with a general-purpose reasoning system. The procedural view stressed efficiency, whereas the declarative view stressed the fact that the overall internal representation can be decomposed into separate *sentences*, each of which has an identifiable meaning. Advocates of knowledge-based systems often cited the following advantages:

Ease of construction: knowledge-based systems can be constructed simply by encoding domain knowledge extracted from an expert; the system builder need not construct and encode a *solution* to the problems in the domain.

Flexibility: the same knowledge can be used to answer a variety of questions and as a component in a variety of systems; the same reasoning mechanism can be used for all domains.

Modularity: each piece of knowledge can be identified, encoded, and debugged independently of the other pieces.

Learnability: various learning methods exist that can be used to extract the required knowledge from data, whereas it is very hard to construct programs by automatic means.

Explainability: a knowledge-based system can *explain* its decisions by reference to the explicit knowledge it contains.

With arguments such as these, the declarative view prevailed and led to the boom in expert systems in the late 1970s and early 1980s.

Unfortunately for the field, the early knowledge-based systems were seldom equal to the challenges of the real world, and since then there has been a great deal of research to remedy these failings.

The area of KNOWLEDGE REPRESENTATION deals with methods for encoding knowledge in a form that can be processed by a computer to derive consequences. Formal LOGIC is used, in various forms, to represent definite knowledge. To handle areas where definite knowledge is not available—for example, medical diagnosis—methods have been developed for representation and reasoning under UNCERTAINTY, including the extension of logic to so-called NONMONOTONIC LOGICS. All knowledge representation systems need some process for KNOWLEDGE ACQUISITION, and much as been done to automate this process through better interface tools, machine learning methods, and, most recently, extraction from natural language texts. Finally, substantial progress has been made on the question of the computational complexity of reasoning.

Logical representation and reasoning. Logical reasoning is appropriate when the available knowledge is definite. McCarthy's (1958) "Advice Taker" paper proposed first-order logic (FOL) as a formal language for the representation of common-sense knowledge in AI systems. FOL has sufficient expressive power for most purposes, including the representation of objects, relations among objects, and universally quantified statements about sets of objects.

Thanks to work by a long line of philosophers and mathematicians, who were also interested in a formal language for representing general (as well as mathematical) knowledge, FOL came with a well-defined syntax and semantics, as well as the powerful guarantee of *completeness*: there exists a computational procedure such that, if the answer to a question is entailed by the available knowledge, then the procedure will find that answer (see GÖDEL'S THEOREMS). More expressive languages than FOL generally do not allow completeness—roughly put, there exist theorems in these languages that cannot be proved.

The first complete LOGICAL REASONING SYSTEM for FOL, the resolution method, was devised by Robinson (1965). An intense period of activity followed in which logical reasoning systems were applied to mathematics, automatic programming, planning, and general-knowledge question answering. Theorem-proving systems for full FOL have proved new theorems in mathematics and have found widespread application in areas such as program verification, which spun off from mainstream AI in the early 1970s.

Despite these early successes, AI researchers soon realized that the computational complexity of general-purpose reasoning with full FOL is prohibitive—such systems could not scale up to handle large knowledge bases. A great deal of attention has therefore been given to more restricted languages. *Database systems*, which have long been distinct from AI, are essentially logical question-answering systems whose knowledge bases are restricted to very simple sentences about specific

objects. *Propositional* languages avoid objects altogether, representing the world by the discrete values of a fixed set of propositional variables and by logical combinations thereof. (Most neural network models fall into this category also.) Propositional reasoning methods based on CONSTRAINT SATISFACTION and GREEDY LOCAL SEARCH have been very successful in real-world applications, but the restricted expressive power of propositional languages severely limits their scope. Much closer to the expressive power of FOL are the languages used in LOGIC PROGRAMMING. While still allowing most kinds of knowledge to expressed very naturally, logic programming systems such as Prolog provide much more efficient reasoning and can work with extremely large knowledge bases.

Reasoning systems must have content with which to reason. Researchers in knowledge representation study methods for codifying and reasoning with particular kinds of knowledge. For example, McCarthy (1963) proposed the SITUATION CALCULUS as a way to represent states of the world and the effects of actions within first-order logic. Early versions of the situation calculus suffered from the the infamous FRAME PROBLEM—the apparent need to specify sentences in the knowledge base for all the *non-effects* of actions. Some philosophers see the frame problem as evidence of the impossibility of the formal, knowledge-based approach to AI, but simple technical advances have in fact resolved the original issues.

Situation calculus is perhaps the simplest form of TEMPORAL REASONING; other formalisms have been developed that provide substantially more general frameworks for handling time and extended events. Reasoning about knowledge itself is important particularly when dealing with other agents, and is usually handled by MODAL LOGIC, an extension of FOL. Other topics studied include reasoning about ownership and transactions, reasoning about substances (as distinct from objects), and reasoning about physical representations of information. A general *ontology*—literally, a description of existence—ties all these areas together into a unified taxonomic hierarchy of categories. FRAME-BASED SYSTEMS are often used to represent such hierarchies, and use specialized reasoning methods based on *inheritance* of properties in the hierarchy.

Logical decision making. An agent's job is to make *decisions*, i.e., to commit to particular actions. The connection between logical reasoning and decision making is simple: the agent must conclude, based on its knowledge, that a certain action is best. In philosophy, this is known as *practical reasoning*. There are many routes to such conclusions. The simplest leads to a form of reactive system using *condition-action rules* of the form "If P then do A." Somewhat more complex reasoning is required when the agent has explicitly represented *goals*. A goal G is a description of

a desired state of affairs—for example, one might have the goal "On vacation in the Seychelles." The *practical syllogism*, first expounded by Aristotle, says that if G is a goal, and A achieves G, then A should be done. Obviously, this rule is open to many objections: it does not specify which of many eligible As should be done, nor does it account for possibly disastrous side-effects of A. Nonetheless, it underlies most forms of decision-making in the logical context.

Often, there will be no single action A that achieves the goal G, but a solution may exist in the form of a *sequence* of actions. Finding such a sequence is called PROBLEM SOLVING, where the word "problem" refers to a task defined by a set of actions, an initial state, a goal, and a set of reachable states. Much of the early cognitive modeling work of NEWELL and SIMON (1972) focused on problem solving, which was seen as a quintessentially intelligent activity. A great deal of research has been done on efficient algorithms for problem solving in the areas of HEURISTIC SEARCH and GAME-PLAYING. The "cognitive structure" of such systems is very simple, and problem-solving competence is often achieved by means of searching through huge numbers of possibilities. For example, the Deep Blue chess program, which recently defeated human world champion Gary Kasparov, often examined over a billion positions prior to each move. Human competence is not thought to involve such computations (see CHESS, PSYCHOLOGY OF).

Most problem-solving algorithms treat the states of the world as atomic—that is, the internal structure of the state representation is not accessible to the algorithm as it considers the possible sequences of actions. This fails to take advantage of two very important sources of power for intelligent systems: the ability to *decompose* complex problems into subproblems and the ability to identify relevant actions from explicit goal descriptions. For example, an intelligent system should be able decompose the goal "have groceries and a clean car" into the subgoals "have groceries" and "have a clean car." Furthermore, it should immediately consider buying groceries and washing the car. Most search algorithms, on the other hand, may consider a variety of action sequences—sitting down, standing up, going to sleep, etc.—before happening on some actions that are relevant.

In principle, a logical reasoning system using McCarthy's situation calculus can generate the kinds of reasoning behaviors necessary for decomposing complex goals and selecting relevant actions. For reasons of computational efficiency, however, special-purpose PLANNING systems have been developed, originating with the STRIPS planner used by Shakey the Robot (Fikes & Nilsson, 1971). Modern planners have been applied to logistical problems that are, in some cases, too complex for humans to handle effectively.

Representation and reasoning under uncertainty. In many areas to which one might wish to apply knowledge-based systems, the available knowledge is far from definite. For example, a person who experiences recurrent headaches may suffer from migraines or a brain tumor. A logical reasoning system can represent this sort of disjunctive information, but cannot represent or reason with the belief that migraine is a *more likely* explanation. Such reasoning is obviously essential for DIAGNOSIS, and has turned out to be central for expert systems in almost all areas. The theory of *probability* (see FOUNDATIONS OF PROBABILITY) is now widely accepted as the basic calculus for reasoning under uncertainty (but see FUZZY LOGIC for a complementary view). Questions remain as to whether it is a good model for human reasoning (see articles on TVERSKY and PROBABILISTIC REASONING), but within AI many of the computational and representational problems that deterred early researchers have been resolved. The adoption of a probabilistic approach has also created rich connections with statistics and control theory.

Standard probability theory views the world as comprised of a set of interrelated random variables whose values are initially unknown. Knowledge comes in the form of *prior* probability distributions over the possible assignments of values to subsets of the random variables. Then, when evidence is obtained about the values of some of the variables, inference algorithms can infer *posterior* probabilities for the remaining unknown variables. Early attempts to use probabilistic reasoning in AI came up against complexity barriers very soon, because the number of probabilities that make up the prior probability distribution can grow exponentially in the number of variables considered.

Starting in the early 1980s, researchers in AI, decision analysis, and statistics developed what are now known as BAYESIAN NETWORKS (Pearl, 1988). These networks give structure to probabilistic knowledge bases by expressing *conditional independence* relationships among the variables. For example, given the actual temperature, the temperature measurements of two thermometers are independent. In this way, Bayesian networks capture our intuitive notions of the causal structure of the domain of application. In most cases, the number of probabilities that must be specified in a Bayesian network grows only linearly with the number of variables. Such systems can therefore handle quite large problems, and applications are very widespread. Moreover, methods exist for *learning* Bayesian networks from raw data (see BAYESIAN LEARNING), making them a natural bridge between the symbolic and neural-network approaches to AI.

In earlier sections, we have stressed the importance of the distinction between propositional and first-order languages. So far, probability theory has been limited to essentially propositional representations; this prevents its application to the more complex forms of cognition addressed by first-order methods. The attempt to unify probability theory and first-order logic, two of the

most fundamental developments in the history of mathematics and philosophy, is among the more important topics in current AI research.

Decision making under uncertainty. Just as logical reasoning is connected to action through goals, probabilistic reasoning is connected to action through *utilities*, which describe an agent's preferences for some states over others. It is a fundamental result of UTILITY THEORY (see also RATIONAL CHOICE THEORY) that an agent whose preferences obey certain rationality constraints, such as transitivity, can be modeled as possessing a *utility function* that assigns a numerical value to each possible state. Furthermore, RATIONAL DECISION MAKING (see also RATIONAL AGENCY) consists of selecting an action to maximize the expected utility of outcome states. An agent that makes rational decisions will, on average, do better than an agent that does not—at least as far as satisfying its own preferences is concerned.

In addition their fundamental contributions to utility theory, Von Neumann and Morgenstern (1944) also developed GAME THEORY to handle the case where the environment contains other agents, which must be modeled as independent utility maximizers. In some game-theoretic situations, it can be shown that optimal behavior must be *randomized*. Additional complexities arise when dealing with so-called *sequential* decision problems, which are analogous to planning problems in the logical case. DYNAMIC PROGRAMMING algorithms, developed in the field of operations research, can generate optimal behavior for such problems. (See also the discussion of REINFORCEMENT LEARNING below.)

In a sense, the theory of rational decision making provides a zeroth-order theory of intelligence, since it provides an operational definition of what an agent *ought* to do in any situation. Virtually every problem an agent faces, including such problems as how to gather information and how to update its beliefs given that information, can be formulated within the theory and, in principle, solved. What the theory ignores is the question of complexity, which we discuss in the final section of this introduction.

Learning. Learning has been a central aspect of AI from its earliest days. It is immediately apparent that learning is a vital characteristic of any intelligent system that has to deal with changing environments. Learning may also be the only way in which complex and competent systems can be constructed—a proposal stated clearly by Turing (1950), who devoted a quarter of his paper to the topic. Perhaps the first major public success for AI was Arthur Samuel's (1959) checker-playing system, which learned to play checkers to a level far superior to its creator's abilities

and attracted substantial television coverage. State-of-the-art systems in almost all areas of AI now use learning to avoid the need for the system designer to have to anticipate and provide knowledge to handle every possible contingency. In some cases, e.g., speech recognition, humans are simply incapable of providing the necessary knowledge accurately.

The discipline of MACHINE LEARNING has become perhaps the largest subfield of AI well as a meeting point between AI and various other engineering disciplines concerned with the design of autonomous, robust systems. (See also the article on the psychology of LEARNING.) An enormous variety of learning systems has been studied in the AI literature, but once superficial differences are stripped away, there seem to be a few core principles at work. To reveal these principles it helps to classify a given learning system along a number of dimensions: 1) the type of feedback available, 2) the component of the agent to be improved, 3) how that component is represented, and 4) the role of prior knowledge. It is also important to be aware that there is a tradeoff between learning and inference and different systems rely more on one than on the other.

The type of feedback available is perhaps the most useful categorizer of learning algorithms. Broadly speaking, learning algorithms fall into the categories of *supervised learning*, *unsupervised learning*, and *reinforcement learning*. Supervised learning algorithms (see, e.g., DECISION TREES and SUPERVISED LEARNING IN MULTILAYER NEURAL NETWORKS) require that a target output is available for every input, an assumption that is natural in some situations (e.g., categorization problems with labeled data, imitation problems, and prediction problems, in which the present can be used as a target for a prediction based on the past). UNSUPERVISED LEARNING algorithms simply find structure in an ensemble of data, whether or not this structure is useful for a particular classification or prediction (examples include clustering algorithms, dimensionality-reducing algorithms, and algorithms that find independent components). REINFORCEMENT LEARNING algorithms require an evaluation signal that gives some measure of progress without necessarily providing an example of correct behavior. Reinforcement learning research has had a particular focus on temporal learning problems, in which the evaluation arrives after a sequence of responses.

The different components of an agent generally have different kinds of representational and inferential requirements. Sensory and motor systems must interface with the physical world and therefore generally require continuous representations and smooth input-output behavior. In such situations, NEURAL NETWORKS have provided a useful class of architectures, as have probabilistic systems such as HIDDEN MARKOV MODELS and BAYESIAN NETWORKS. The latter models also are generally characterized by a clear propositional semantics, and as such have been exploited for elementary cognitive processing. DECISION TREES are also propositional systems

that are appropriate for simple cognitive tasks. There are variants of decision trees that utilize continuous representations, and these have close links with neural networks, as well as variants of decision trees that utilize relational machinery, making a connection with INDUCTIVE LOGIC PROGRAMMING. The latter class of architecture provides the full power of first-order logic and the capability of learning complex symbolic theories.

Prior knowledge is an important component of essentially all modern learning architectures, particularly so in architectures that involve expressive representations. Indeed, the spirit of inductive logic programming is to use the power of logical inference to bootstrap background knowledge and to interpret new data in the light of that knowledge. This approach is carried to what is perhaps its (logical) extreme in the case of EXPLANATION-BASED LEARNING, in which the system derives a new datum from its current theory—learning in this case can be viewed as a sophisticated form of caching.

Underlying all research on learning is a version of the general problem of INDUCTION; in particular, on what basis can we expect that a system that performs well on past "training" data should also perform well on future "test" data? The theory of learning (see COMPUTATIONAL LEARNING THEORY and STATISTICAL LEARNING THEORY) attacks this problem by assuming that the data provided to a learner is obtained from a fixed but unknown probability distribution. The theory yields a notion of *sample complexity*, which quantifies the amount of data that a learner must see in order to expect–with high probability–to perform (nearly) as well in the future as in the past. The theory also provides support for the intuitive notion of Ockham's razor—the idea that if a simple hypothesis performs as well as a complex hypothesis, one should prefer the simple hypothesis.

General ideas from probability theory, in the form of BAYESIAN LEARNING, and related ideas from INFORMATION THEORY, in the form of the MINIMUM DESCRIPTION LENGTH approach, provide a link between learning theory and learning practice. In particular, Bayesian learning, which views learning as the updating of probabilistic beliefs in hypotheses given evidence, naturally embodies a form of Ockham's razor. Bayesian methods have been applied to neural networks, Bayesian networks, decision trees, and many other learning architectures.

We have seen that learning has strong relationships to knowledge representation and to the study of uncertainty. There are also important connections between learning and search. In particular, most learning algorithms involve some form of search through the hypothesis space to find hypotheses that are consistent (or nearly so) with the data and with prior expectations. Standard heuristic search algorithms are often invoked—either explicitly or implicitly—to perform this search. EVO-

LUTIONARY COMPUTATION also treats learning as a search process, in which the "hypothesis" is an entire agent, and learning takes place by "mutation" and "natural selection" of agents that perform well. There are also interesting links between learning and planning; in particular, it is possible to view REINFORCEMENT LEARNING as a form of "on-line" planning.

Finally, it is worth noting that learning has been a particularly successful branch of AI research in terms of its applications to real-world problems in specific fields; see for example the articles on PATTERN RECOGNITION AND LAYERED NETWORKS, STATISTICAL TECHNIQUES IN NLP, VISION AND LEARNING, and ROBOTICS AND LEARNING.

Language. NATURAL LANGUAGE PROCESSING, or NLP—the ability to perceive, understand, and generate language—is an essential part of HUMAN-COMPUTER INTERACTION as well as the most obvious task to be solved in passing the Turing test. As with logical reasoning, AI researchers have benefited from a pre-existing intellectual tradition. The field of LINGUISTICS (see also LINGUISTICS, PHILOSOPHICAL ISSUES) has produced formal notions of syntax and semantics, the view of utterances as *speech acts*, and very careful philosophical analyses of the meanings of various constructs in natural language. The field of COMPUTATIONAL LINGUISTICS has grown up since the 1960s as a fertile union of ideas from AI, cognitive science, and linguistics.

As soon as programs were written to process natural language, it became obvious that the problem was much harder than had been anticipated. Substantial effort was devoted in the U.S. to Russian–English translation from 1957 onwards, but in 1966 a government report concluded that "There has been no machine translation of general scientific text, and none is in immediate prospect." Successful translation appeared to require an *understanding* of the content of the text; the barriers included massive ambiguity (both syntactic and semantic), a huge variety of word senses, and the vast numbers of idiosyncratic ways of using words to convey meanings. Overcoming these barriers seems to require the use of large amounts of common-sense knowledge and the ability to reason with it—in other words, solving a large fraction of the AI problem. For this reason, Robert Wilensky has described natural language processing as an "AI-complete" problem (see also MODULARITY AND LANGUAGE).

Research in NLP has uncovered a great deal of new information about language. There is a better appreciation of the *actual* syntax of natural language—as opposed to the vastly oversimplified models that held sway before computational investigation was possible. Several new families of FORMAL GRAMMARS have been proposed as a result. In the area of semantics, dozens of interesting phenomena have surfaced—for example, the surprising range of semantic relationships in

noun–noun pairs such as "alligator shoes" and "baby shoes." In the area of discourse understanding, researchers have found that grammaticality is sometimes thrown out of the window, leading some to propose that grammar itself is not a useful construct for NLP.

One consequence of the richness of natural language is that it is very difficult to build by hand a system capable of handling anything close to the full range of phenomena. Most systems constructed prior to the 1990s functioned only in predefined and highly circumscribed domains. Stimulated in part by the availability of large online text corpora, the use of STATISTICAL TECHNIQUES IN NATURAL LANGUAGE PROCESSING has created something of a revolution. Instead of building complex grammars by hand, these techniques train very large but very simple probabilistic grammars and semantic models from millions of words of text. These techniques have reached the point where they can be usefully applied to extract information from general newspaper articles.

Few researchers expect simple probability models to yield human-level understanding. On the other hand, the view of language entailed by this approach—that the text is a form of *evidence* from which higher-level facts can be inferred by a process of probabilistic inference—may prove crucial for further progress in NLP. A probabilistic framework allows the smooth integration of the multiple "cues" required for NLP, such as syntax, semantics, discourse conventions, and prior expectations.

In contrast to the general problem of natural language understanding, the problem of SPEECH RECOGNITION IN MACHINES may be feasible without recourse to general knowledge and reasoning capabilities. The statistical approach was taken much earlier in the speech field, beginning in the mid-1970s. Together with improvements in the signal processing methods used to extract acoustic features, this has led to steady improvements in performance, to the point where commercial systems can handle dictated speech with over 95% accuracy. The combination of speech recognition and SPEECH SYNTHESIS promises to make interaction with computers much more natural for humans. Unfortunately, accuracy rates for natural dialogue seldom exceed 75%; possibly, speech systems will have to rely on knowledge-based expectations and real understanding to make further progress.

Vision. The study of vision presents a number of advantages—visual processing systems are present across a wide variety of species, they are reasonably accessible experimentally (psychophysically, neuropsychologically, and neurophysiologically), and a wide variety of artificial imaging systems are available that are sufficiently similar to their natural counterparts so as to make research in machine vision highly relevant to research in natural vision. An integrated view of the prob-

lem has emerged, linking research in COMPUTATIONAL VISION, which is concerned with the development of explicit theories of human and animal vision, with MACHINE VISION, which is concerned with the development of an engineering science of vision.

Computational approaches to vision, including the influential theoretical framework of MARR, generally involve a succession of processes that begin with localized numeric operations on images (so-called "early vision") and proceed towards the high-level abstractions thought to be involved in object recognition. The current view is that the interpretation of complex scenes involves inference in both the bottom-up and top-down directions.

High-level object recognition is not the only purpose of vision. Representations at intermediate levels can also be an end unto themselves, directly subserving control processes of orienting, locomotion, reaching, and grasping. Visual analysis at all levels can be viewed as a process of recovering aspects of the visual scene from its projection onto a 2-D image. Visual properties such as shape and TEXTURE behave in lawful ways under the geometry of perspective projection, and understanding this geometry has been a focus of research. Related geometrical issues have been studied in STEREO and MOTION PERCEPTION, where additionally the issue of finding correspondences between multiple images arises. In all of these cases, localized spatial and temporal cues are generally highly ambiguous with respect to the aspects of the scene from which they arise, and algorithms that recover such aspects generally involve some form of spatial and/or temporal integration.

It is also important to prevent integrative processes from wrongly smoothing across discontinuities that correspond to visually meaningful boundaries. Thus, visual processing also requires segmentation. A wide variety of algorithms have been studied for the segmentation of image data. Again, an understanding of projective geometry has been a guide for the development of such algorithms.

Integration and segmentation are also required at higher levels of visual processing, where more abstract principles (such as those studied by the Gestalt psychologists; see GESTALT PERCEPTION) are needed to group visual elements.

Finally, in many cases the goal of visual processing is to detect or recognize objects in the visual scene. A number of difficult issues arise in VISUAL OBJECT RECOGNITION, include the issue of what kinds of features should be used (2-D or 3-D, edge-based or filter-based), how to deal with missing features (e.g., due to occlusion or shadows), how to represent flexible objects (such as humans), and how to deal with variations in pose and lighting. Methods based on learning (cf. VISION AND LEARNING) have played an increasingly important role in addressing some of these issues.

Robotics. Robotics is the control of physical effectors to achieve physical tasks such as navigation and assembly of complex objects. Effectors include grippers and arms to perform MANIPULATION AND GRASPING and wheels and legs for MOBILE ROBOTS and WALKING/RUNNING MACHINES.

The need to interact directly with a physical environment, which is generally only partially known and partially controllable, brings certain issues to the fore in robotics that are often skirted in other areas in AI. One important set of issues arises from the fact that environments are generally dynamical systems, characterizable by a large (perhaps infinite) collection of real-valued state variables, whose values are not generally directly observable by the robot (i.e., they are "hidden"). The presence of the robot control algorithm itself as a feedback loop in the environment introduces additional dynamics. The robot designer must be concerned with the issue of *stability* in such a situation. Achieving stability not only prevents disasters but it also simplifies the dynamics, providing a degree of predictability that is essential for the success of planning algorithms.

Stability is a key issue in manipulation and grasping, where the robot must impart a distributed pattern of forces and torques to an object so as to maintain a desired position and orientation in the presence of external disturbances (such as gravity). Research has tended to focus on static stability (ignoring the dynamics of the grasped object). Static stability is also of concern in the design of walking and running robots, although rather more pertinent is the problem of dynamic stability, in which a moving robot is stabilized by taking advantage of its inertial dynamics.

Another important set of issues in robotics has to do with UNCERTAINTY. Robots are generally equipped with a limited set of sensors and these sensors are generally noisy and inherently ambiguous. To a certain extent the issue is the same as that treated in the discussion of vision above, and the solutions, involving algorithms for integration and smoothing, are often essentially the same. In robotics, however, the sensory analysis is generally used to subserve a control law and the exigencies of feedback control introduce new problems (cf. CONTROL THEORY). Processing time must be held to a minimum and the system must focus on obtaining only that information needed for control. These objectives can be difficult to meet, and recent research in robotics has focused on minimizing the need for feedback, designing sequences of control actions that are guaranteed to bring objects into desired positions and orientations regardless of the initial conditions.

Uncertainty is due not only to noisy sensors and hidden states, but also to ignorance about the structure of the environment. Many robot systems actively model the environment, using system identification techniques from control theory, as well as more general supervised and unsupervised methods from MACHINE LEARNING. Specialized representations are often used to represent

obstacles ("configuration space") and location in space (graphs and grids). Probabilistic approaches are often used to explicitly represent and manipulate uncertainty within these formalisms.

In classical robotic control methodology, the system attempts to recover as much of the state of the environment as possible, operates on the internal representation of the state using general planning and reasoning algorithms, and chooses a sequence of control actions to implement the selected plan. The sheer complexity of designing this kind of architecture has led researchers to investigate simpler architectures that make do with minimal internal state. BEHAVIOR-BASED ROBOTICS approaches the problem via an interacting set of elemental processes called "behaviors," each of which is a simplified control law relating sensations and actions. REINFORCEMENT LEARNING has provided algorithms that utilize simplified evaluation signals to guide a search for improved laws; over time these algorithms approach the optimal plans that are derived (with more computational effort) from explicit planning algorithms (see LEARNING AND ROBOTICS).

Complexity, Rationality, and Intelligence We have observed at several points in this introduction that COMPUTATIONAL COMPLEXITY is a major problem for intelligent agents. To the extent that they can be analyzed, most of the problems of perceiving, learning, reasoning, and decision making are believed to have a worst-case complexity that is at least exponential in the size of the problem description. Exponential complexity means that, for example, a problem of size 100 would take 10 billion years to solve on the fastest available computers. Given that humans face much larger problems than this all the time—we receive as input several billion bytes of information every second—one wonders how we manage at all.

Of course, there are a number of mitigating factors: an intelligent agent must deal largely with the typical case, not the worst case, and accumulated experience with similar problems can greatly reduce the difficulty of new problems. The fact remains, however, that humans cannot even come close to achieving perfectly rational behavior—most of us do fairly poorly even on problems such as chess, which is an *infinitesimal* subset of the real world. What, then, is the right thing for an agent to do, if it cannot possibly compute the right thing to do?

In practical applications of AI, one possibility is to restrict the allowable set of problems to those that are efficiently soluble. For example, deductive database systems use restricted subsets of logic that allow for polynomial-time inference. Such research has given us a much deeper understanding of the sources of complexity in reasoning, but does not seem directly applicable to the problem of general intelligence. Somehow, we must face up to the inevitable compromises that must be made in the quality of decisions that an intelligent agent can make. Descriptive theo-

ries of such compromises—for example, Herbert Simon's work on *satisficing*—appeared soon after the development of formal theories of rationality. Normative theories of BOUNDED RATIONALITY address the question at the end of the preceding paragraph by examining what is achievable with fixed computational resources. One promising approach is to devote some of those resources to METAREASONING (see also METACOGNITION), i.e., reasoning about what reasoning to do. The technique of EXPLANATION-BASED LEARNING (a formalization of the common psychological concept of *chunking* or *knowledge compilation*) helps an agent cope with complexity by caching efficient solutions to common problems. REINFORCEMENT LEARNING methods enable an agent to learn effective (if not perfect) behaviors in complex environments without the need for extended problem-solving computations.

What is interesting about all these aspects of intelligence is that without the need for effective use of limited computational resources, they make no sense. That is, computational complexity may be responsible for many, perhaps most, of the aspects of cognition that make intelligence an interesting subject of study. In contrast, the cognitive structure of an infinitely powerful computational device could be very straightforward indeed.

Additional sources. 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 & Nilsson, 1981). Early papers on neural networks are collected in *Neurocomputing* (Anderson & Rosenfeld, 1988). The *Encyclopedia of AI* (Shapiro, 1992) contains survey articles on almost every topic in AI. The four-volume *Handbook of Artificial Intelligence* (Barr & Feigenbaum, 1981) contains descriptions of almost every major AI system published before 1981. Standard texts on AI include *Artificial Intelligence: A Modern Approach* (Russell & Norvig, 1995) and *Artificial Intelligence: A New Synthesis* (Nilsson, 1998). Historical surveys include Kurzweil (1990) and Crevier (1993).

The most recent work appears in the proceedings of the major AI conferences: the biennial International Joint Conference on AI (IJCAI); the annual National Conference on AI, more often known as AAAI after its sponsoring organization; and the European Conference on AI (ECAI). 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, some of which are listed in the relevant articles. The main professional societies for AI are the American Associa-

tion 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.

References

Anderson, J. A., & Rosenfeld, E. (Eds.). (1988). *Neurocomputing: Foundations of Research.* MIT Press, Cambridge, Massachusetts.

Barr, A., Cohen, P. R., & Feigenbaum, E. A. (Eds.). (1989). *The Handbook of Artificial Intelligence*, Vol. 4. Addison-Wesley, Reading, Massachusetts.

Barr, A., & Feigenbaum, E. A. (Eds.). (1981). *The Handbook of Artificial Intelligence*, Vol. 1. HeurisTech Press and William Kaufmann, Stanford, California and Los Altos, California. First of four volumes; other volumes published separately (Barr & Feigenbaum, 1982; Cohen & Feigenbaum, 1982; Barr, Cohen, & Feigenbaum, 1989).

Barr, A., & Feigenbaum, E. A. (Eds.). (1982). *The Handbook of Artificial Intelligence*, Vol. 2. HeurisTech Press and William Kaufmann, Stanford, California and Los Altos, California.

Boden, M. A. (1977). *Artificial Intelligence and Natural Man.* Basic Books, New York.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, *47*(1–3), 139–159.

Chomsky, N. (1957). *Syntactic Structures.* Mouton, The Hague and Paris.

Cohen, P. R., & Feigenbaum, E. A. (Eds.). (1982). *The Handbook of Artificial Intelligence*, Vol. 3. HeurisTech Press and William Kaufmann, Stanford, California and Los Altos, California.

Crevier, D. (1993). *AI: The Tumultuous History of the Search for Artificial Intelligence.* Basic Books, New York.

Dreyfus, H. L. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason.* MIT Press, Cambridge, Massachusetts.

Feigenbaum, E. A., & Feldman, J. (Eds.). (1963). *Computers and Thought.* McGraw-Hill, New York.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, *2*(3–4), 189–208.

Haugeland, J. (Ed.). (1985). *Artificial Intelligence: The Very Idea*. MIT Press, Cambridge, Massachusetts.

Kurzweil, R. (1990). *The Age of Intelligent Machines*. MIT Press, Cambridge, Massachusetts.

McCarthy, J. (1958). Programs with common sense. In *Proceedings of the Symposium on Mechanisation of Thought Processes*, Vol. 1, pp. 77–84 London. Her Majesty's Stationery Office.

McCarthy, J. (1963). Situations, actions, and causal laws. Memo 2, Stanford University Artificial Intelligence Project, Stanford, California.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–137.

Minsky, M. L. (Ed.). (1968). *Semantic Information Processing*. MIT Press, Cambridge, Massachusetts.

Minsky, M. L. (1986). *The society of mind*. Simon and Schuster, New York.

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, New Jersey.

Nilsson, N. J. (1998). *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann, San Mateo, California.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.

Poole, D., Mackworth, A., & Goebel, R. (1997). *Computational intelligence: A logical approach*. Oxford University Press, Oxford.

Raphael, B. (1976). *The Thinking Computer: Mind Inside Matter*. W. H. Freeman, New York.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, *12*, 23–41.

Russell, S. J., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Englewood Cliffs, New Jersey.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210–229.

Shapiro, S. C. (Ed.). (1992). *Encyclopedia of Artificial Intelligence* (second edition). Wiley, New York.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*, 433—460.

Von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior* (first edition). Princeton University Press, Princeton, New Jersey.

Webber, B. L., & Nilsson, N. J. (Eds.). (1981). *Readings in Artificial Intelligence.* Morgan Kaufmann, San Mateo, California.

Wellman, M. P. (1990). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, *44*(3), 257–303.