2

Clues

Clues about what might be needed to make machines intelligent are scattered abundantly throughout philosophy, logic, biology, psychology, statistics, and engineering. With gradually increasing intensity, people set about to exploit clues from these areas in their separate quests to automate some aspects of intelligence. I begin my story by describing some of these clues and how they inspired some of the first achievements in artificial intelligence.

2.1 From Philosophy and Logic

Although people had reasoned logically for millennia, it was the Greek philosopher Aristotle who first tried to analyze and codify the process. Aristotle identified a type of reasoning he called the *syllogism* "... in which, certain things being stated, something other than what is stated follows of necessity from their being so."[1]

Here is a famous example of one kind of syllogism:[2]

1. *All humans are mortal.* (stated)
2. *All Greeks are humans.* (stated)
3. *All Greeks are mortal.* (result)

The beauty (and importance for AI) of Aristotle's contribution has to do with the *form* of the syllogism. We aren't restricted to talking about humans, Greeks, or mortality. We could just as well be talking about something else – a result made obvious if we rewrite the syllogism using arbitrary symbols in the place of *humans*, *Greeks*, and *mortal*. Rewriting in this way would produce

1. *All B's are A.* (stated)
2. *All C's are B's.* (stated)
3. *All C's are A.* (result)

One can substitute anything one likes for *A*, *B*, and *C*. For example, *all athletes are healthy* and *all soccer players are athletes*, and therefore *all soccer players are healthy*, and so on. (Of course, the "result" won't necessarily be true unless the things "stated" are. Garbage in, garbage out!)

Aristotle's logic provides two clues to how one might automate reasoning. First, patterns of reasoning, such as syllogisms, can be economically represented as *forms* or *templates*. These use generic symbols, which can stand for many different concrete instances. Because they can stand for anything, the symbols themselves are unimportant.

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

Figure 2.1. Gottfried Leibniz.

Second, after the general symbols are replaced by ones pertaining to a specific problem, one only has to "turn the crank" to get an answer. The use of general symbols and similar kinds of crank-turning are at the heart of all modern AI reasoning programs.

In more modern times, Gottfried Wilhelm Leibniz (1646–1716; Fig. 2.1) was among the first to think about logical reasoning. Leibniz was a German philosopher, mathematician, and logician who, among other things, co-invented the calculus. (He had lots of arguments with Isaac Newton about that.) But more importantly for our story, he wanted to mechanize reasoning. Leibniz wrote[3]

> It is unworthy of excellent men to lose hours like slaves in the labor of calculation which could safely be regulated to anyone else if machines were used.

and

> For if praise is given to the men who have determined the number of regular solids . . . how much better will it be to bring under mathematical laws human reasoning, which is the most excellent and useful thing we have.

Leibniz conceived of and attempted to design a language in which all human knowledge could be formulated – even philosophical and metaphysical knowledge. He speculated that the propositions that constitute knowledge could be built from a smaller number of primitive ones – just as all words can be built from letters in an alphabetic language. His *lingua characteristica* or universal language would consist of these primitive propositions, which would comprise an *alphabet for human thoughts*.

The alphabet would serve as the basis for automatic reasoning. His idea was that if the items in the alphabet were represented by numbers, then a complex proposition could be obtained from its primitive constituents by multiplying the corresponding numbers together. Further arithmetic operations could then be used to determine whether or not the complex proposition was true or false. This whole process was to be accomplished by a *calculus ratiocinator* (calculus of reasoning). Then, when

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

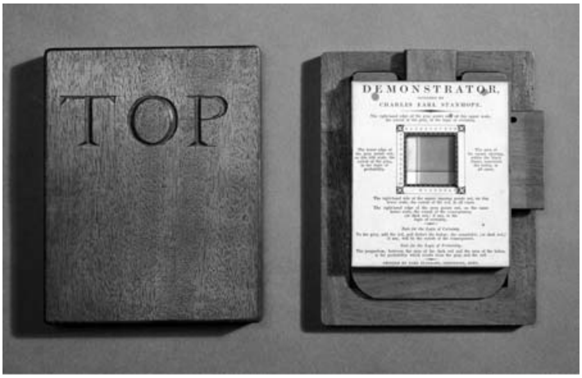

Figure 2.2. The Stanhope Square Demonstrator, 1805. (Photograph courtesy of Science Museum/SSPL.)

philosophers disagreed over some problem they could say, "*calculemus*" ("let us calculate"). They would first pose the problem in the *lingua characteristica* and then solve it by "turning the crank" on the *calculus ratiocinator*.

The main problem in applying this idea was discovering the components of the primitive "alphabet." However, Leibniz's work provided important additional clues to how reasoning might be mechanized: Invent an alphabet of simple symbols and the means for combining them into more complex expressions.

Toward the end of the eighteenth century and the beginning of the nineteenth, a British scientist and politician, Charles Stanhope (Third Earl of Stanhope), built and experimented with devices for solving simple problems in logic and probability. (See Fig. 2.2.) One version of his "box" had slots on the sides into which a person could push colored slides. From a window on the top, one could view slides that were appropriately positioned to represent a specific problem. Today, we would say that Stanhope's box was a kind of analog computer.

The book *Computing Before Computers* gives an example of its operation:[4]

> To solve a numerical syllogism, for example:
>
> *Eight of ten A's are B's*; *Four of ten A's are C's*;
> *Therefore, at least two B's are C's*.
>
> Stanhope would push the red slide (representing B) eight units across the window (representing A) and the gray slide (representing C) four units from the opposite direction. The two units that the slides overlapped represented the minimum number of B's that were also C's.
> . . .
> In a similar way, the Demonstrator could be used to solve a traditional syllogism like:
>
> *No M is A*; *All B is M*; *Therefore, No B is A*.

Stanhope was rather secretive about his device and didn't want anyone to know what he was up to. As mentioned in *Computing Before Computers*, "The few friends and relatives who received his privately distributed account of the Demonstrator, *The Science of Reasoning Clearly Explained Upon New Principles* (1800), were advised

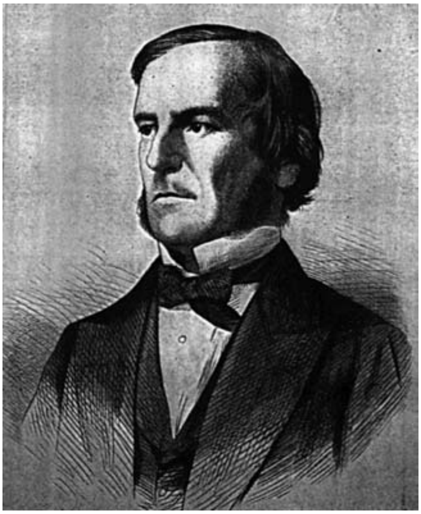

Figure 2.3. George Boole.

to remain silent lest ‘some bastard imitation’ precede his intended publication on the subject.”

But no publication appeared until sixty years after Stanhope’s death. Then, the Reverend Robert Harley gained access to Stanhope’s notes and one of his boxes and published an article on what he called “The Stanhope Demonstrator.”[5]

Contrasted with Llull’s schemes and Leibniz’s hopes, Stanhope built the first logic machine that actually worked – albeit on small problems. Perhaps his work raised confidence that logical reasoning could indeed be mechanized.

In 1854, the Englishman George Boole (1815–1864; Fig. 2.3) published a book with the title *An Investigation of the Laws of Thought on Which Are Founded the Mathematical Theories of Logic and Probabilities*.[6] Boole’s purpose was (among other things) “to collect . . . some probable intimations concerning the nature and constitution of the human mind.” Boole considered various logical principles of human reasoning and represented them in mathematical form. For example, his “Proposition IV” states “. . . *the principle of contradiction . . . affirms that it is impossible for any being to possess a quality, and at the same time not to possess it. . . .*” Boole then wrote this principle as an algebraic equation,

$$x(1-x)=0,$$

in which x represents “any class of objects,” $(1-x)$ represents the “contrary or supplementary class of objects,” and 0 represents a class that “does not exist.”

In Boolean algebra, an outgrowth of Boole’s work, we would say that 0 represents *falsehood*, and 1 represents *truth*. Two of the fundamental operations in logic, namely OR and AND, are represented in Boolean algebra by the operations $+$ and $\times$, respectively. Thus, for example, to represent the statement “either p or q or both,” we would write $p+q$. To represent the statement “p and q,” we would write $p \times q$. Each of p and q could be true or false, so we would evaluate the value (truth

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

or falsity) of $p+q$ and $p \times q$ by using definitions for how $+$ and $\times$ are used, namely,

$$\begin{aligned}1+0&=1,\\1\times 0&=0,\\1+1&=1,\\1\times 1&=1,\\0+0&=0,\end{aligned}$$

and

$$0 \times 0 = 0.$$

Boolean algebra plays an important role in the design of telephone switching circuits and computers. Although Boole probably could not have envisioned computers, he did realize the importance of his work. In a letter dated January 2, 1851, to George Thomson (later Lord Kelvin) he wrote[7]

> I am now about to set seriously to work upon preparing for the press an account of my theory of Logic and Probabilities which in its present state I look upon as the most valuable if not the only valuable contribution that I have made or am likely to make to Science and the thing by which I would desire if at all to be remembered hereafter . . .

Boole's work showed that some kinds of logical reasoning could be performed by manipulating equations representing logical propositions – a very important clue about the mechanization of reasoning. An essentially equivalent, but not algebraic, system for manipulating and evaluating propositions is called the "propositional calculus" (often called "propositional logic"), which, as we shall see, plays a very important role in artificial intelligence. [Some claim that the Greek Stoic philospher Chrysippus (280–209 BCE) invented an early form of the propositional calculus.[8]]

One shortcoming of Boole's logical system, however, was that his propositions p, q, and so on were "atomic." They don't reveal any entities *internal* to propositions. For example, if we expressed the proposition "Jack is human" by p, and "Jack is mortal" by q, there is nothing in p or q to indicate that the Jack who is human is the very same Jack who is mortal. For that, we need, so to speak, "molecular expressions" that have internal elements.

Toward the end of the nineteenth century, the German mathematician, logician, and philosopher Friedrich Ludwig Gottlob Frege (1848–1925) invented a system in which propositions, along with their internal components, could be written down in a kind of graphical form. He called his language *Begriffsschrift*, which can be translated as "concept writing." For example, the statement "All persons are mortal" would have been written in *Begriffsschrift* something like the diagram in Fig. 2.4.[9]

Note that the illustration explicitly represents the x who is predicated to be a person and that it is the same x who is then claimed to be mortal. It's more convenient nowadays for us to represent this statement in the linear form $(\forall x)P(x) \supset M(x)$, whose English equivalent is "for all x, if x is a person, then x is mortal."

Frege's system was the forerunner of what we now call the "predicate calculus," another important system in artificial intelligence. It also foreshadows another

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

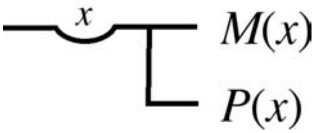

Figure 2.4. Expressing “All persons are mortal” in *Begriffsschrift*.

representational form used in present-day artificial intelligence: semantic networks. Frege’s work provided yet more clues about how to mechanize reasoning processes. At last, sentences expressing information to be reasoned about could be written in unambiguous, symbolic form.

2.2 From Life Itself

In Proverbs 6:6–8, King Solomon says “Go to the ant, thou sluggard; consider her ways and be wise.” Although his advice was meant to warn against slothfulness, it can just as appropriately enjoin us to seek clues from biology about how to build or improve artifacts.

Several aspects of “life” have, in fact, provided important clues about intelligence. Because it is the *brain* of an animal that is responsible for converting sensory information into action, it is to be expected that several good ideas can be found in the work of neurophysiologists and neuroanatomists who study brains and their fundamental components, neurons. Other ideas are provided by the work of psychologists who study (in various ways) intelligent behavior as it is actually happening. And because, after all, it is evolutionary processes that have produced intelligent life, those processes too provide important hints about how to proceed.

2.2.1 *Neurons and the Brain*

In the late nineteenth and early twentieth centuries, the “neuron doctrine” specified that living cells called “neurons” together with their interconnections were fundamental to what the brain does. One of the people responsible for this suggestion was the Spanish neuroanatomist Santiago Ramón y Cajal (1852–1934). Cajal (Fig. 2.5) and Camillo Golgi won the Nobel Prize in Physiology or Medicine in 1906 for their work on the structure of the nervous system.

A neuron is a living cell, and the human brain has about ten billion (10^{10}) of them. Although they come in different forms, typically they consist of a central part called a *soma* or *cell body*, incoming fibers called *dendrites*, and one or more outgoing fibers called *axons*. The axon of one neuron has projections called *terminal buttons* that come very close to one or more of the dendrites of other neurons. The gap between the terminal button of one neuron and a dendrite of another is called a *synapse*. The size of the gap is about 20 nanometers. Two neurons are illustrated schematically in Fig. 2.6.

Through electrochemical action, a neuron may send out a stream of pulses down its axon. When a pulse arrives at the synapse adjacent to a dendrite of another neuron, it may act to excite or to inhibit electrochemical activity of the other neuron across

Figure 2.5. Ramón y Cajal.

the synapse. Whether or not this second neuron then "fires" and sends out pulses of its own depends on how many and what kinds of pulses (excitatory or inhibitory) arrive at the synapses of its various incoming dendrites and on the efficiency of those synapses in transmitting electrochemical activity. It is estimated that there are over half a trillion synapses in the human brain. The neuron doctrine claims that the

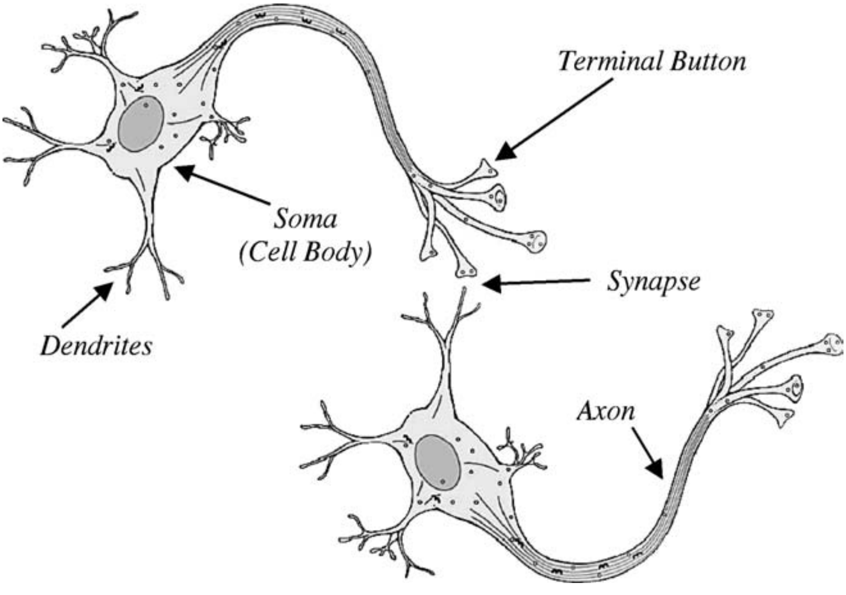

Figure 2.6. Two neurons. (Adapted from *Science*, Vol. 316, p. 1416, 8 June 2007. Used with permission.)

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

Figure 2.7. Warren McCulloch.

various activities of the brain, including perception and thinking, are the result of all of this neural activity.

In 1943, the American neurophysiologist Warren McCulloch (1899–1969; Fig. 2.7) and logician Walter Pitts (1923–1969) claimed that the neuron was, in essence, a "logic unit." In a famous and important paper they proposed simple models of neurons and showed that networks of these models could perform all possible computational operations.[10] The McCulloch–Pitts "neuron" was a mathematical abstraction with inputs and outputs (corresponding, roughly, to dendrites and axons, respectively). Each output can have the value 1 or 0. (To avoid confusing a McCulloch–Pitts neuron with a real neuron, I'll call the McCulloch–Pitts version, and others like it, a "neural element.") The neural elements can be connected together into networks such that the output of one neural element is an input to others and so on. Some neural elements are excitatory – their outputs contribute to "firing" any neural elements to which they are connected. Others are inhibitory – their outputs contribute to inhibiting the firing of neural elements to which they are connected. If the sum of the excitatory inputs less the sum of the inhibitory inputs impinging on a neural element is greater than a certain "threshold," that neural element fires, sending its output of 1 to all of the neural elements to which it is connected.

Some examples of networks proposed by McCullough and Pitts are shown in Fig. 2.8.

The Canadian neuropsychologist Donald O. Hebb (1904–1985) also believed that neurons in the brain were the basic units of thought. In an influential book,[11] Hebb suggested that "when an axon of cell *A* is near enough to excite *B* and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that *A*'s efficiency, as one of the cells firing *B*, is

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

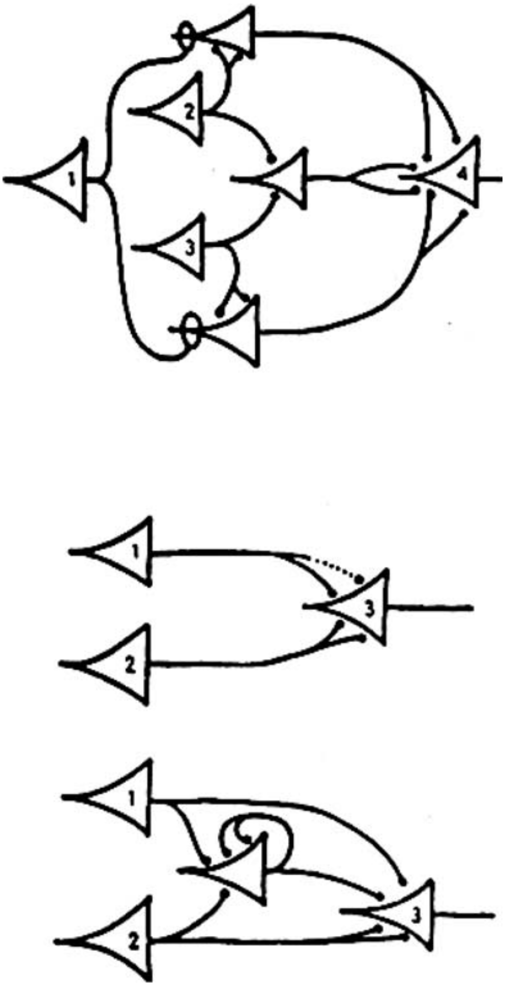

Figure 2.8. Networks of McCulloch–Pitts neural elements. (Adapted from Fig. 1 of Warren S. McCulloch and Walter Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133, 1943.)

increased." Later, this so-called Hebb rule of change in neural "synaptic strength" was actually observed in experiments with living animals. (In 1965, the neurophysiologist Eric Kandel published results showing that simple forms of learning were associated with synaptic changes in the marine mollusk *Aplysia californica*. In 2000, Kandel shared the Nobel Prize in Physiology or Medicine "for their discoveries concerning signal transduction in the nervous system.")

Hebb also postulated that groups of neurons that tend to fire together formed what he called *cell assemblies*. Hebb thought that the phenomenon of "firing together" tended to persist in the brain and was the brain's way of representing the perceptual event that led to a cell-assembly's formation. Hebb said that "thinking" was the sequential activation of sets of cell assemblies.[12]

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

Figure 2.9. B. F. Skinner. (Courtesy of the B. F. Skinner Foundation.)

2.2.2 *Psychology and Cognitive Science*

Psychology is the science that studies mental processes and behavior. The word is derived from the Greek words *psyche*, meaning breath, spirit, or soul, and *logos*, meaning word. One might expect that such a science would have much to say of interest to those wanting to create intelligent artifacts. However, until the late nineteenth century, most psychological theorizing depended on the insights of philosophers, writers, and other astute observers of the human scene. (Shakespeare, Tolstoy, and other authors were no slouches when it came to understanding human behavior.)

Most people regard serious scientific study to have begun with the German Wilhelm Wundt (1832–1920) and the American William James (1842–1910).[13] Both established psychology labs in 1875 – Wundt in Leipzig and James at Harvard. According to C. George Boeree, who teaches the history of psychology at Shippensburg University in Pennsylvania, "The method that Wundt developed is a sort of experimental introspection: The researcher was to carefully observe some simple event – one that could be measured as to quality, intensity, or duration – and record his responses to variations of those events." Although James is now regarded mainly as a philosopher, he is famous for his two-volume book *The Principles of Psychology*, published in 1873 and 1874.

Both Wundt and James attempted to say something about *how* the brain worked instead of merely cataloging its input–output behavior. The psychiatrist Sigmund Freud (1856–1939) went further, postulating internal components of the brain, namely, the *id*, the *ego*, and the *superego*, and how they interacted to affect behavior. He thought one could learn about these components through his unique style of guided introspection called *psychoanalysis*.

Attempting to make psychology more scientific and less dependent on subjective introspection, a number of psychologists, most famously B. F. Skinner (1904–1990; Fig. 2.9), began to concentrate solely on what could be objectively measured, namely, specific behavior in reaction to specific stimuli. The *behaviorists* argued that psychology should be a science of behavior, not of the mind. They rejected the idea

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

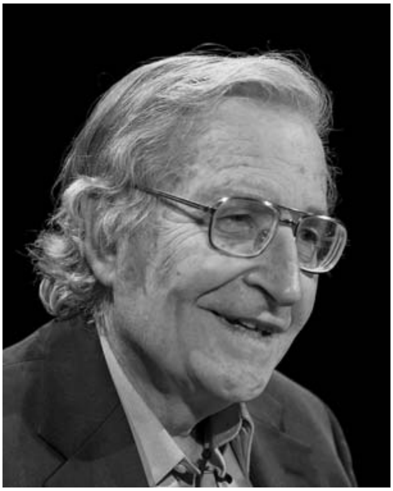

Figure 2.10. Noam Chomsky. (Photograph by Don J. Usner.)

of trying to identify internal mental states such as beliefs, intentions, desires, and goals.

This development might at first be regarded as a step backward for people wanting to get useful clues about the internal workings of the brain. In criticizing the statistically oriented theories arising from "behaviorism," Marvin Minsky wrote "Originally intended to avoid the need for 'meaning,' [these theories] manage finally only to avoid the possibility of explaining it."[14] Skinner's work did, however, provide the idea of a *reinforcing* stimulus – one that rewards recent behavior and tends to make it more likely to occur (under similar circumstances) in the future.

Reinforcement learning has become a popular strategy among AI researchers, although it does depend on internal states. Russell Kirsch (circa 1930–), a computer scientist at the U.S. National Bureau of Standards (now the National Institute for Standards and Technology, NIST), was one of the first to use it. He proposed how an "artificial animal" might use reinforcement to learn good moves in a game. In some 1954 seminar notes he wrote the following:[15] "The animal model notes, for each stimulus, what move the opponent next makes, . . . Then, the next time that same stimulus occurs, the animal duplicates the move of the opponent that followed the same stimulus previously. The more the opponent repeats the same move after any given stimulus, the more the animal model becomes 'conditioned' to that move."

Skinner believed that reinforcement learning could even be used to explain verbal behavior in humans. He set forth these ideas in his 1957 book *Verbal Behavior*,[16] claiming that the laboratory-based principles of selection by consequences can be extended to account for what people say, write, gesture, and think.

Arguing against Skinner's ideas about language the linguist Noam Chomsky (1928– ; Fig. 2.10), in a review[17] of Skinner's book, wrote that

> careful study of this book (and of the research on which it draws) reveals, however, that [Skinner's] astonishing claims are far from justified . . . the insights that have been achieved in the laboratories of the reinforcement theorist, though quite genuine, can be applied to complex human behavior only in the most gross and superficial way, and that speculative

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

> attempts to discuss linguistic behavior in these terms alone omit from consideration factors of fundamental importance . . .

How, Chomsky seems to ask, can a person produce a potentially infinite variety of previously unheard and unspoken sentences having arbitrarily complex structure (as indeed they can do) through experience alone? These "factors of fundamental importance" that Skinner omits are, according to Chomsky, linguistic abilities that must be innate – not learned. He suggested that "human beings are somehow specially created to do this, with data-handling or 'hypothesis-formulating' ability of [as yet] unknown character and complexity." Chomsky claimed that all humans have at birth a "universal grammar" (or developmental mechanisms for creating one) that accounts for much of their ability to learn and use languages.[18]

Continuing the focus on internal mental processes and their limitations, the psychologist George A. Miller (1920–) analyzed the work of several experimenters and concluded that the "immediate memory" capacity of humans was approximately seven "chunks" of information.[19] In the introduction to his paper about this "magical number," Miller humorously notes "My problem is that I have been persecuted by an integer. For seven years this number has followed me around, has intruded in my most private data, and has assaulted me from the pages of our most public journals. This number assumes a variety of disguises, being sometimes a little larger and sometimes a little smaller than usual, but never changing so much as to be unrecognizable. The persistence with which this number plagues me is far more than a random accident." Importantly, he also claimed that "the span of immediate memory seems to be almost independent of the number of bits per chunk." That is, it doesn't matter what a chunk represents, be it a single digit in a phone number, a name of a person just mentioned, or a song title; we can apparently only hold seven of them (plus or minus two) in our immediate memory.

Miller's paper on "The Magical Number Seven," was given at a Symposium on Information Theory held from September 10 to 12, 1956, at MIT.[20] Chomsky presented an important paper there too. It was entitled "Three Models for the Description of Language," and in it he proposed a family of rules of syntax he called *phrase-structure grammars*.[21] It happens that two pioneers in AI research (of whom we'll hear a lot more later), Allen Newell (1927–1992), then a scientist at the Rand Corporation, and Herbert Simon (1916–2001), a professor at the Carnegie Institute of Technology (now Carnegie Mellon University), gave a paper there also on a computer program that could prove theorems in propositional logic. This symposium, bringing together as it did scientists with these sorts of overlapping interests, is thought to have contributed to the birth of *cognitive science*, a new discipline devoted to the study of the mind. Indeed, George Miller wrote[22]

> I went away from the Symposium with a strong conviction, more intuitive than rational, that human experimental psychology, theoretical linguistics, and computer simulation of cognitive processes were all pieces of a larger whole, and that the future would see progressive elaboration and coordination of their shared concerns . . .

In 1960, Miller and colleagues wrote a book proposing a specific internal mechanism responsible for behavior, which they called the TOTE unit (Test–Operate–Test–Exit).[23] There is a TOTE unit corresponding to every goal that an agent

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

might have. Using its perceptual abilities, the unit first tests whether or not its goal is satisfied. If so, the unit rests (exits). If not, some operation specific to achieving that goal is performed, the test for goal achievement is performed again, and so on repetitively until the goal finally is achieved. As a simple example, consider the TOTE unit for driving a nail with a hammer. So long as the nail is not completely driven in (the goal), the hammer is used to strike it (the operation). Pounding stops (the exit) when the goal is finally achieved. It's difficult to say whether or not this book inspired similar work by artificial intelligence researchers. The idea was apparently "in the air," because at about the same time, as we shall see later, some early work in AI used very similar ideas. [I can say that my work at SRI with behavior (intermediate-level) programs for the robot, Shakey, and my later work on what I called "teleo-reactive" programs were influenced by Miller's ideas.]

Cognitive science attempted to explicate internal mental processes using ideas such as goals, memory, task queues, and strategies without (at least during its beginning years) necessarily trying to ground these processes in neurophysiology.[24] Cognitive science and artificial intelligence have been closely related ever since their beginnings. Cognitive science has provided clues for AI researchers, and AI has helped cognitive science with newly invented concepts useful for understanding the workings of the mind.

2.2.3 *Evolution*

That living things evolve gives us two more clues about how to build intelligent artifacts. First, and most ambitiously, the processes of evolution itself – namely, random generation and selective survival – might be simulated on computers to produce the machines we dream about. Second, those paths that evolution followed in producing increasingly intelligent animals can be used as a guide for creating increasingly intelligent artifacts. Start by simulating animals with simple tropisms and proceed along these paths to simulating more complex ones. Both of these strategies have been followed with zest by AI researchers, as we shall see in the following chapters. Here, it will suffice to name just a few initial efforts.

Early attempts to simulate evolution on a computer were undertaken at Princeton's Institute for Advanced Study by the viral geneticist Nils Aall Barricelli (1912–1993). His 1954 paper described experiments in which numbers migrated and reproduced in a grid.[25]

Motivated by the success of biological evolution in producing complex organisms, some researchers began thinking about how programs could be evolved rather than written. R. N. Friedberg and his IBM colleagues[26] conducted experiments in which, beginning with a population of random computer programs, they attempted to evolve ones that were more successful at performing a simple logical task. In the summary of his 1958 paper, Friedberg wrote that "[m]achines would be more useful if they could learn to perform tasks for which they were not given precise methods. . . . It is proposed that the program of a stored-program computer be gradually improved by a learning procedure which tries many programs and chooses, from the instructions that may occupy a given location, the one most often associated with a successful result." That is, Friedberg installed instructions from "successful" programs into the

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

programs of the next "generation," much as how the genes of individuals successful enough to have descendants are installed in those descendants.

Unfortunately, Friedberg's attempts to evolve programs were not very successful. As Marvin Minsky pointed out,[27]

> The machine [described in the first paper] did learn to solve some extremely simple problems. But it took of the order of 1000 times longer than pure chance would expect....
>
> The second paper goes on to discuss a sequence of modifications... With these, and with some 'priming' (starting the machine off on the right track with some useful instructions), the system came to be only a little worse than chance.

Minsky attributes the poor performance of Friedberg's methods to the fact that each descendant machine differed very little from its parent, whereas any helpful improvement would require a much larger step in the "space" of possible machines.

Other early work on artificial evolution was more successful. Lawrence Fogel (1928–2007) and colleagues were able to evolve machines that could make predictions of the next element in a sequence.[28] Woodrow W. Bledsoe (1921–1995) at Panoramic Research and Hans J. Bremermann (1926–1969) at the University of California, Berkeley, used simulated evolution to solve optimization and mathematical problems, respectively.[29] And Ingo Rechenberg (according to one AI researcher) "pioneered the method of artificial evolution to solve complex optimization tasks, such as the design of optimal airplane wings or combustion chambers of rocket nozzles."[30]

The first prominent work inspired by biological evolution was John Holland's development of "genetic algorithms" beginning in the early 1960s. Holland (1929–), a professor at the University of Michigan, used strings of binary symbols (0's and 1's), which he called "chromosomes" in analogy with the genetic material of biological organisms. (Holland says he first came up with the notion while browsing through the Michigan math library's open stacks in the early 1950s.)[31] The encoding of 0's and 1's in a chromosome could be interpreted as a solution to some given problem. The idea was to evolve chromosomes that were better and better at solving the problem. Populations of chromosomes were subjected to an evolutionary process in which individual chromosomes underwent "mutations" (changing a component 1 to a 0 and vice versa), and pairs of the most successful chromosomes at each stage of evolution were combined to make a new chromosome. Ultimately, the process would produce a population containing a chromosome (or chromosomes) that solved the problem.[32]

Researchers would ultimately come to recognize that all of these evolutionary methods were elaborations of a very useful mathematical search strategy called "gradient ascent" or "hill climbing." In these methods, one searches for a local maximum of some function by taking the steepest possible uphill steps. (When searching for a local minimum, the analogous method is called "gradient descent.")

Rather than attempt to duplicate evolution itself, some researchers preferred to build machines that followed along evolution's paths toward intelligent life. In the late 1940s and early 1950s, W. Grey Walter (1910–1977), a British neurophysiologist (born in Kansas City, Missouri), built some machines that behaved like some of life's most primitive creatures. They were wheeled vehicles to which he gave the

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

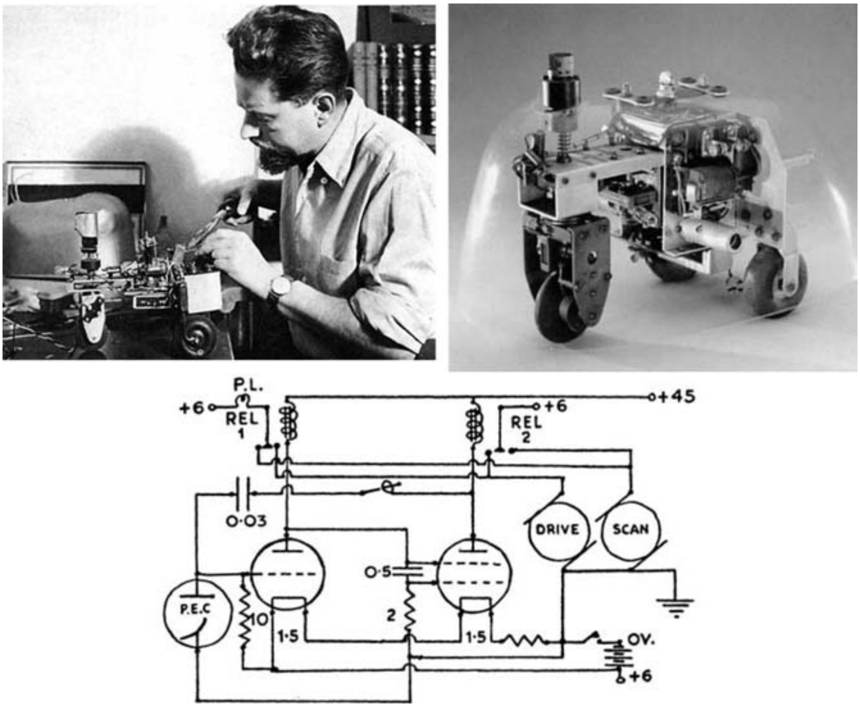

Figure 2.11. Grey Walter (top left), his *Machina speculatrix* (top right), and its circuit diagram (bottom). (Grey Walter photograph from Hans Moravec, *ROBOT*, Chapter 2: Caution! Robot Vehicle!, p. 18, Oxford: Oxford University Press, 1998; "Turtle" photograph courtesy of National Museum of American History, Smithsonian Institution; the circuit diagram is from W. Grey Walter, *The Living Brain*, p. 200, London: Gerald Duckworth & Co., Ltd., 1953.)

taxonomic name *Machina speculatrix* (machine that looks; see Fig. 2.11).[33] These tortoise-like machines were controlled by "brains" consisting of very simple vacuum-tube circuits that sensed their environments with photocells and that controlled their wheel motors. The circuits could be arranged so that a machine either moved toward or away from a light mounted on a sister machine. Their behaviors seemed purposive and often complex and unpredictable, so much so that Walter said they "might be accepted as evidence of some degree of self-awareness." *Machina speculatrix* was the beginning of a long line of increasingly sophisticated "behaving machines" developed by subsequent researchers.

2.2.4 *Development and Maturation*

Perhaps there are alternatives to rerunning evolution itself or to following its paths toward increasing complexity from the most primitive animals. By careful study of the behavior of young children, the Swiss psychologist Jean Piaget proposed a set of stages in the maturation of their thinking abilities from infancy to adolescence.[34] Might these stages provide a set of steps that could guide designers of intelligent artifacts? Start with a machine that is able to do what an infant can do, and then design machines that can mimic the abilities of children at each rung of the ladder. This strategy might be called "ontogenetic" to contrast it with the "phylogenetic" strategy of using simlulated evolution.

Of course, it may be that an infant mind is far too complicated to simulate and the processes of its maturation too difficult to follow. In any case, this particular clue remains to be exploited.

2.2.5 *Bionics*

At a symposium in 1960, Major Jack E. Steele, of the Aerospace Division of the United States Air Force, used the term "bionics" to describe the field that learns lessons from nature to apply to technology.[35]

Several bionics and bionics-related meetings were held during the 1960s. At the 1963 Bionics Symposium, Leonard Butsch and Hans Oestreicher wrote "Bionics aims to take advantage of millions of years of evolution of living systems during which they adapted themselves for optimum survival. One of the outstanding successes of evolution is the information processing capability of living systems [the study of which is] one of the principal areas of Bionics research."[36]

Today, the word "bionics" is concerned mainly with orthotic and prosthetic devices, such as artificial cochleas, retinas, and limbs. Nevertheless, as AI researchers continue their quest, the study of living things, their evolution, and their development may continue to provide useful clues for building intelligent artifacts.

2.3 From Engineering

2.3.1 *Automata, Sensing, and Feedback*

Machines that move by themselves and even do useful things by themselves have been around for centuries. Perhaps the most common early examples are the "verge-and-foliot" weight-driven clocks. (See Fig. 2.12.) These first appeared in the late Middle Ages in the towers of large Italian cities. The verge-and-foliot mechanism converted the energy of a falling weight into stepped rotational motion, which could be used to move the clock hands. Similar mechanisms were elaborated to control the actions of automata, such as those of the Munich Glockenspiel.

One of the first automatic machines for producing goods was Joseph-Marie Jacquard's weaving loom, built in 1804. (See Fig. 2.13.) It followed a long history of looms and improved on the "punched card" design of Jacques de Vaucanson's loom of 1745. (Vaucanson did more than build mechanical ducks.) The punched cards of the Jacquard loom controlled the actions of the shuttles, allowing automatic production of fabric designs. Just a few years after its invention, there were some 10,000 Jacquard looms weaving away in France. The idea of using holes in paper or cards was later adopted by Herman Hollerith for tabulating the 1890 American census data and in player pianos (using perforated rolls instead of cards). The very first factory "robots" of the so-called pick-and-place variety used only modest elaborations of this idea.

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

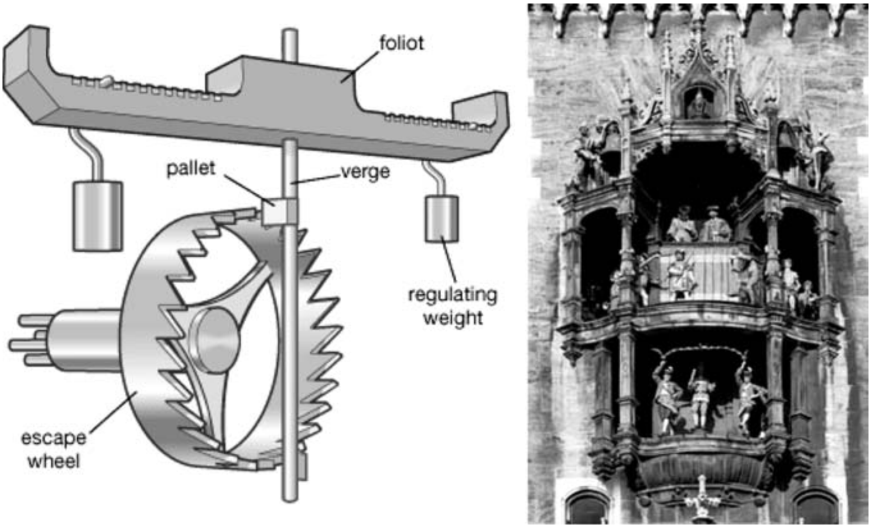

Figure 2.12. A verge-and-foliot mechanism (left) and automata at the Munich Glockenspiel (right).

It was only necessary to provide these early machines with an external source of energy (a falling weight, a wound-up spring, or humans pumping pedals). Their behavior was otherwise fully automatic, requiring no human guidance. But, they had an important limitation – they did not perceive anything about their environments. (The punched cards that were "read" by the Jacquard loom are considered part of the machine – not part of the environment.) Sensing the environment and then letting what is sensed influence what a machine does is critical to intelligent behavior. Grey Walters's "tortoises," for example, had photocells that could detect the presence or

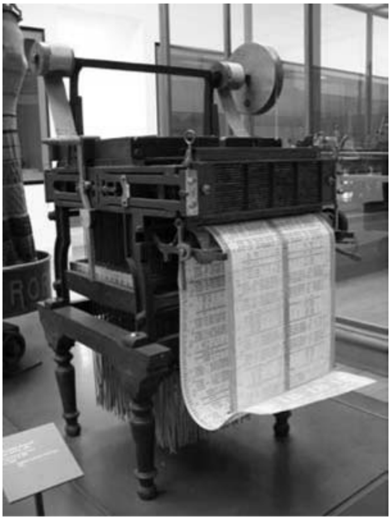

Figure 2.13. Reconstruction of a Jacquard loom.

absence of light in their environments and act accordingly. Thus, they seem more intelligent than a Jacquard loom or clockwork automata.

One of the simplest ways to allow what is sensed to influence behavior involves what is called "feedback control." The word derives from feeding some aspect of a machine's behavior, say its speed of operation, back into the internals of the machine. If the aspect of behavior that is fed back acts to diminish or reverse that aspect, the process is called "negative feedback." If, on the other hand, it acts to increase or accentuate that aspect of behavior, it is called "positive feedback." Both types of feedback play extremely important roles in engineering.

Negative feedback techniques have been used for centuries in mechanical devices. In 270 BCE, a Greek inventor and barber, Ktesibios of Alexandria, invented a float regulator to keep the water level in a tank feeding a water clock at a constant depth by controlling the water flow into the tank.[37] The feedback device was a float valve consisting of a cork at the end of a rod. The cork floated on the water in the tank. When the water level in the tank rose, the cork would rise, causing the rod to turn off the water coming in. When the water level fell, the cork would fall, causing the rod to turn on the water. The water level in modern flush toilets is regulated in much the same way. In 250 BCE, Philon of Byzantium used a similar float regulator to keep a constant level of oil in a lamp.[38]

The English clockmaker John Harrison (1693–1776) used a type of negative feedback control in his clocks. The ambient temperature of a clock affects the length of its balance spring and thus its time-keeping accuracy. Harrison used a bimetallic strip (sometimes a rod), whose curvature depends on temperature. The strip was connected to the balance spring in such a way that it produced offsetting changes in the length of the spring, thus making the clock more independent of its temperature. The strip senses the temperature and causes the clock to behave differently, and more accurately, than it otherwise would. Today, such bimetallic strips see many uses, notably in thermostats. (Dava Sobel's 1995 book, *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time*, recounts the history of Harrison's efforts to build a prize-winning clock for accurate time-keeping at sea.)

Perhaps the most graphic use of feedback control is the centrifugal flyball governor perfected in 1788 by James Watt for regulating the speed of his steam engine. (See Fig. 2.14.) As the speed of the engine increases, the balls fly outward, which causes a linking mechanism to decrease air flow, which causes the speed to decrease, which causes the balls to fall back inward, which causes the speed to increase, and so on, resulting in an equilibrium speed.

In the early 1940s, Norbert Wiener (1894–1964) and other scientists noted similarities between the properties of feedback control systems in machines and in animals. In particular, inappropriately applied feedback in control circuits led to jerky movements of the system being controlled that were similar to pathological "tremor" in human patients. Arturo Rosenblueth, Norbert Wiener, and Julian Bigelow coined the term "cybernetics" in a 1943 paper. Wiener's book by that name was published in 1948. The word is related to the word "governor." (In Latin *gubernaculum* means helm, and *gubernator* means helmsman. The Latin derives from the Greek *kybernetike*, which means the art of steersmanship.[39])

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

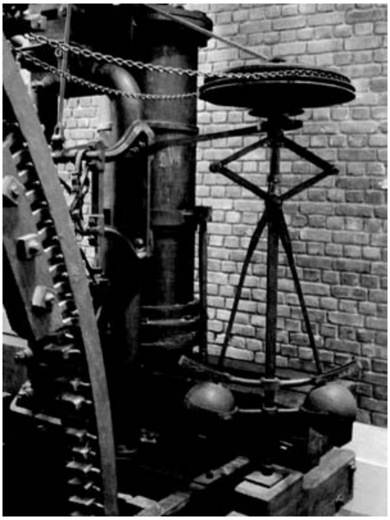

Figure 2.14. Watt's flyball governor.

Today, the prefix "cyber" is used to describe almost anything that deals with computers, robots, the Internet, and advanced simulation. For example, the author William Gibson coined the term "cyberspace" in his 1984 science fiction novel *Neuromancer*. Technically, however, cybernetics continues to describe activities related to feedback and control.[40]

The English psychiatrist W. Ross Ashby (1903–1972; Fig. 2.15) contributed to the field of cybernetics by his study of "ultrastability" and "homeostasis." According to Ashby, ultrastability is the capacity of a system to reach a stable state under a wide variety of environmental conditions. To illustrate the idea, he built an electromechanical device called the "homeostat." It consisted of four pivoted magnets whose positions were rendered interdependent through feedback mechanisms. If the position of any was disturbed, the effects on the others and then back on itself would result in all of them returning to an equilibrium condition. Ashby described this device in Chapter 8 of his influential 1952 book *Design For a Brain*. His ideas had an influence on several AI researchers. My "teleo-reactive programs," to be described later, were motivated in part by the idea of homeostasis.

Another source of ideas, loosely associated with cybernetics and bionics, came from studies of "self-organizing systems." Many unorganized combinations of simple parts, including combinations of atoms and molecules, respond to energetic "jostling" by falling into stable states in which the parts are organized in more complex assemblies. An online dictionary devoted to cybernetics and systems theory has a nice example: "A chain made out of paper clips suggests that someone has taken the trouble to link paper clips together to make a chain. It is not in the nature of paper clips to make themselves up into a chain. But, if you take a number of paper clips, open them up slightly and then shake them all together in a cocktail shaker, you will find at the end that the clips have organized themselves into short or long chains. The chains are not so neat as chains put together by hand but, nevertheless, they are chains."[41]

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

Figure 2.15. W. Ross Ashby, Warren McCulloch, Grey Walter, and Norbert Wiener at a meeting in Paris. (From P. de Latil, *Thinking by Machine*, 1956.)

The term "self-organizing" seems to have been first introduced by Ashby in 1947.[42] Ashby emphasized that self-organization is not a property of an organism itself, in response to its environment and experience, but a property of the organism and its environment *taken together*. Although self-organization appears to be important in ideas about how life originated, it is unclear whether or not it provides clues for building intelligent machines.

2.3.2 *Statistics and Probability*

Because nearly all reasoning and decision making take place in the presence of uncertainty, dealing with uncertainty plays an important role in the automation of intelligence. Attempts to quantify uncertainty and "the laws of chance" gave rise to statistics and probability theory. What would turn out to be one of the most important results in probability theory, at least for artificial intelligence, is Bayes's rule, which I'll define presently in the context of an example. The rule is named for Reverend Thomas Bayes (1702–1761), an English clergyman.[43]

One of the important applications of Bayes's rule is in signal detection. Let's suppose a radio receiver is tuned to a station that after midnight broadcasts (randomly) one of two tones, either tone A or tone B, and on a particular night we want to decide which one is being broadcast. On any given day, we do not know ahead of time which tone is to be broadcast that night, but suppose we do know their probabilities. (For example, it might be that both tones are equally probable.) Can

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

we find out which tone is being broadcast by listening to the signal coming into the receiver? Well, listening can't completely resolve the matter because the station is far away, and random noise partially obscures the tone. However, depending on the nature of the obscuring noise, we can often calculate the probability that the actual tone that night is A (or that it is B). Let's call the signal y and the actual tone x (which can be either A or B). The probability that $x = A$, given the evidence for it contained in the incoming signal, y, is written as $p(x = A \mid y)$ and read as "the probability that x is A, given that the signal is y." The probability that $x = B$, given the same evidence is $p(x = B \mid y)$.

A reasonable "decision rule" would be to decide in favor of tone A if $p(x = A \mid y)$ is larger than $p(x = B \mid y)$. Otherwise, decide in favor of tone B. (There is a straightforward adjustment to this rule that takes into account differences in the "costs" of the two possible errors.) The problem in applying this rule is that these two probabilities are not readily calculable, and that is where Bayes's rule comes in. It allows us to calculate these probabilities in terms of other probabilities that are more easily guessed or otherwise obtainable. Specifically, Bayes's rule is

$$p(x \mid y) = p(y \mid x)p(x)/p(y).$$

Using Bayes's rule, our decision rule can now be reformulated as

Decide in favor of tone A if $p(y \mid x = A)p(x = A)/p(y)$ is greater than $p(y \mid x = B)p(x = B)/p(y)$. Otherwise, decide in favor of tone B.

Because $p(y)$ occurs in both expressions and therefore does not affect which one is larger, the rule simplifies to

Decide in favor of tone A if $p(y \mid x = A)p(x = A)$ is greater than $p(y \mid x = B)p(x = B)$. Otherwise, decide in favor of tone B.

We assume that we know the *a priori* probabilities of the tones, namely, $p(x = A)$ and $p(x = B)$, so it remains only for us to calculate $p(y \mid x)$ for $x = A$ and $x = B$. This expression is called the *likelihood* of y given x. When the two probabilities, $p(x = A)$ and $p(x = B)$, are equal (that is, when both tones are equally probable *a priori*), then we can decide in favor of which likelihood is greater. Many decisions that are made in the presence of uncertainty use this "maximum-likelihood" method. The calculation for these likelihoods depends on how we represent the received signal, y, and on the statistics of the interfering noise.

In my example, y is a radio signal, that is, a voltage varying in time. For computational purposes, this time-varying voltage can be represented by a sequence of samples of its values at appropriately chosen, uniformly spaced time points, say $y(t_1)$, $y(t_2), \ldots y(t_i), \ldots, y(t_N)$. When noise alters these values from what they would have been without noise, the probability of the sequence of them (given the cases when the tone is A and when the tone is B) can be calculated by using the known statistical properties of the noise. I won't go into the details here except to say that, for many types of noise statistics, these calculations are quite straightforward.

In the twentieth century, scientists and statisticians such as Karl Pearson (1857–1936), Sir Ronald A. Fisher (1890–1962), Abraham Wald (1902–1950), and Jerzey Neyman (1894–1981) were among those who made important contributions to the

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

use of statistical and probabilistic methods in estimating parameters and in making decisions. Their work set the foundation for some of the first engineering applications of Bayes's rule, such as the one I just illustrated, namely, deciding which, if any, of two or more electrical signals is present in situations where noise acts to obscure the signals. A paper by the American engineers David Van Meter and David Middleton, which I read as a beginning graduate student in 1955, was my own introduction to these applications.[44] For artificial intelligence, these uses of Bayes's rule provided clues about how to mechanize the perception of both speech sounds and visual images. Beyond perception, Bayes's rule lies at the center of much other modern work in artificial intelligence.

2.3.3 *The Computer*

A. Early Computational Devices

Proposals such as those of Leibniz, Boole, and Frege can be thought of as early attempts to provide foundations for what would become the "software" of artificial intelligence. But reasoning and all the other aspects of intelligent behavior require, besides software, some sort of *physical* engine. In humans and other animals, that engine is the brain. The simple devices of Grey Walter and Ross Ashby were, of course, physical manifestations of their ideas. And, as we shall see, early networks of neuron-like units were realized in physical form. However, to explore the ideas inherent in most of the clues from logic, from neurophysiology, and from cognitive science, more powerful engines would be required. While McCulloch, Wiener, Walter, Ashby, and others were speculating about the machinery of intelligence, a very powerful and essential machine bloomed into existence – the general-purpose digital computer. This single machine provided the engine for all of these ideas and more. It is by far the dominant hardware engine for automating intelligence.

Building devices to compute has a long history. William Aspray has edited an excellent book, *Computing Before Computers*, about computing's early days.[45] The first machines were able to do arithmetic calculations, but these were not programmable. Wilhelm Schickard (1592–1635; Fig. 2.16) built one of the first of these in 1623. It is said to have been able to add and subtract six-digit numbers for use in calculating astronomical tables. The machine could "carry" from one digit to the next.

In 1642 Blaise Pascal (1623–1662; Fig. 2.16) created the first of about fifty of his computing machines. It was an adding machine that could perform automatic carries from one position to the next. "The device was contained in a box that was small enough to fit easily on top of a desk or small table. The upper surface of the box . . . consisted of a number of toothed wheels, above which were a series of small windows to show the results. In order to add a number, say 3, to the result register, it was only necessary to insert a small stylus into the toothed wheel at the position marked 3 and rotate the wheel clockwise until the stylus encountered the fixed stop . . . "[46]

Inspired by Pascal's machines, Gottfried Leibniz built a mechanical multiplier called the "Step Reckoner" in 1674. It could add, subtract, and do multiplication

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

Figure 2.16. Wilhelm Schickard (left) and Blaise Pascal (right).

(by repeated additions). “To multiply a number by 5, one simply turned the crank five times.”[47]

Several other calculators were built in the ensuing centuries. A particularly interesting one, which was too complicated to build in its day, was designed in 1822 by Charles Babbage (1791–1871), an English mathematician and inventor. (See Fig. 2.17.) Called the “Difference Engine,” it was to have calculated mathematical tables (of the kind used in navigation at sea, for example) using the method of finite differences. Babbage’s Difference Engine No. 2 was actually constructed in 1991 (using Babbage’s designs and nineteenth-century mechanical tolerances) and is now on display at the London Science Museum. The Museum arranged for another copy

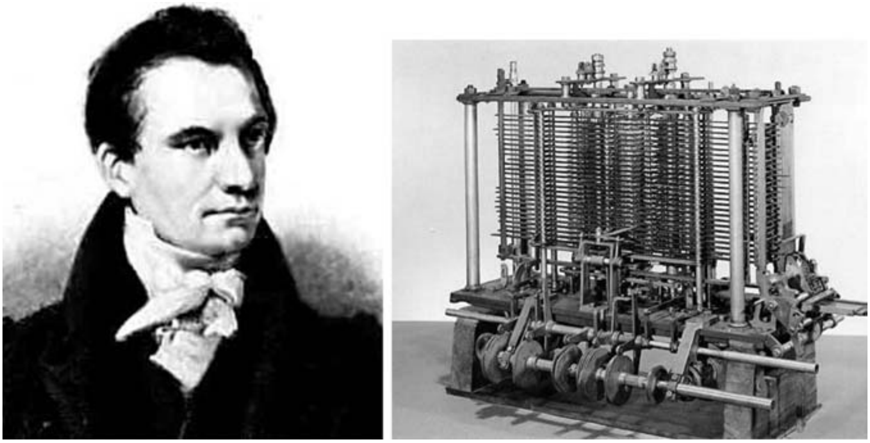

Figure 2.17. Charles Babbage (left) and a model of his Analytical Engine (right).

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

to be built for Nathan Myhrvold, a former Microsoft Chief Technology Officer. (A description of the machine and a movie is available from a Computer History Museum Web page at http://www.computerhistory.org/babbage/.)

Adding machines, however, can only add and subtract (and, by repetition of these operations, also multiply and divide). These are important operations but not the only ones needed. Between 1834 and 1837 Babbage worked on the design of a machine called the "Analytical Engine," which embodied most of the ideas needed for general computation. It could store intermediate results in a "mill," and it could be programmed. However, its proposed realization as a collection of steam-driven, interacting brass gears and cams ran into funding difficulties and was never constructed.

Ada Lovelace (1815–1852), the daughter of Lord Byron, has been called the "world's first programmer" for her alleged role in devising programs for the Analytical Engine. However, in the book *Computing Before Computers* the following claim is made:[48]

> This romantically appealing image is without foundation. All but one of the programs cited in her notes [to her translation of an account of a lecture Babbage gave in Turin, Italy] had been prepared by Babbage from three to seven years earlier. The exception was prepared by Babbage for her, although she did detect a "bug" in it. Not only is there no evidence that Ada Lovelace ever prepared a program for the Analytical Engine but her correspondence with Babbage shows that she did not have the knowledge to do so.

For more information about the Analytical Engine and an emulator and programs for it, see http://www.fourmilab.ch/babbage/.

Practical computers had to await the invention of electrical, rather than brass, devices. The first computers in the early 1940s used electromechanical relays. Vacuum tubes (thermionic valves, as they say in Britain) soon won out because they permitted faster and more reliable computation. Nowadays, computers use billions of tiny transistors arrayed on silicon wafers. Who knows what might someday replace them?

B. Computation Theory

Even before people actually started building computers, several logicians and mathematicians in the 1930s pondered the problem of just what could be computed. Alonzo Church came up with a class of functions that could be computed, ones he called "recursive."[49] The English logician and mathematician, Alan Turing (1912–1954; Fig. 2.18), proposed what is now understood to be an equivalent class – ones that could be computed by an imagined machine he called a "logical computing machine (LCM)," nowadays called a "Turing machine."[50] (See Fig. 2.19.) The claim that these two notions are equivalent is called the "Church–Turing Thesis." (The claim has not been proven, but it is strongly supported by logicians and no counterexample has ever been found.)[51]

The Turing machine is a hypothetical computational device that is quite simple to understand. It consists of just a few parts. There is an infinite tape (which is one reason the device is just imagined and not actually built) divided into cells and a tape drive. Each cell has printed on it either a 1 or a 0. The machine also has a

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

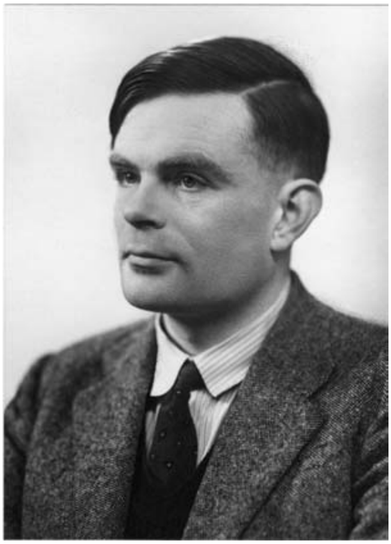

Figure 2.18. Alan Turing. (Alan Mathison Turing by Elliott & Fry © National Portrait Gallery, London)

read–write head positioned over one cell of the tape. The read function reads what is on the tape. There is also a logic unit that can decide, depending on what is read and the state of the logic machine, to change its own state, to command the write function to write either a 1 or a 0 on the cell being read (possibly replacing what is already there), to move the tape one cell to the left or to the right (at which time the new cell is read and so on), or to terminate operation altogether. The input (the "problem" to be computed) is written on the tape initially. (It turns out that any such input can be coded into 1's and 0's.) When, and if, the machine terminates, the output (the coded "answer" to the input problem) ends up being printed on the tape.

Turing proved that one could always specify a particular logic unit (the part that decides on the machine's actions) for his machine such that the machine would

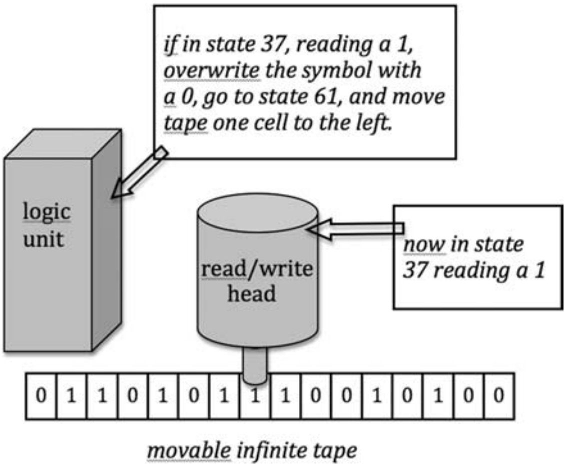
if in state 37, reading a 1,
overwrite the symbol with
a 0, go to state 61, and move
tape one cell to the left.
logic
unit
read/write
head
now in state
37 reading a 1
0 1 1 0 1 0 1 1 1 0 0 1 0 1 0 0
movable infinite tape

Figure 2.19. A Turing machine.

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

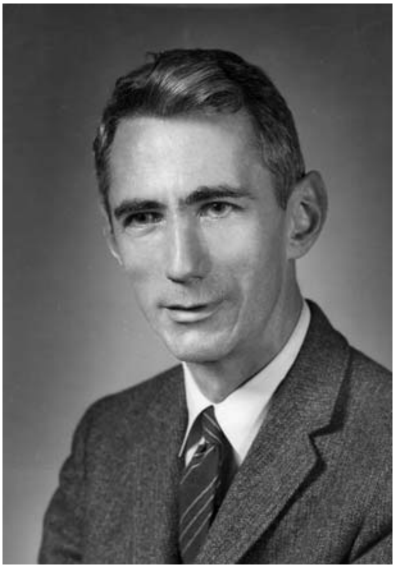

Figure 2.20. Claude Shannon. (Photograph courtesy of MIT Museum.)

compute any computable function. More importantly, he showed that one could encode on the tape itself a prescription for any logic unit specialized for a particular problem and then use a general-purpose logic unit for *all* problems. The encoding for the special-purpose logic unit can be thought of as the "program" for the machine, which is stored on the tape (and thus subject to change by the very operation of the machine!) along with the description of the problem to be solved. In Turing's words, "It can be shown that a single special machine of that type can be made to do the work of all. It could in fact be made to work as a model of any other machine. The special machine may be called the universal machine."[52]

C. Digital Computers

Somewhat independently of Turing, engineers began thinking about how to build actual computing devices consisting of programs and logical circuitry for performing the instructions contained in the programs. Some of the key ideas for designing the logic circuits of computers were developed by the American mathematician and inventor Claude Shannon (1916–2001; Fig. 2.20).[53] In his 1937 Yale University master's thesis[54] Shannon showed that Boolean algebra and binary arithmetic could be used to simplify telephone switching circuits. He also showed that switching circuits (which can be realized either by combinations of relays, vacuum tubes, or whatever) could be used to implement operations in Boolean logic, thus explaining their importance in computer design.

It's hard to know who first thought of the idea of storing a computer's program along with its data in the computer's memory banks. Storing the program allows changes in the program to be made easily, but more importantly it allows the program to change itself by changing appropriate parts of the memory where the program is stored. Among those who might have thought of this idea first are the German engineer Konrad Zuse (1910–1995) and the American computer pioneers J. Presper Eckert (1919–1995) and John W. Mauchly (1907–1980). (Of course Turing had

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

already proposed storing what amounted to a program on the tape of a universal Turing machine.)

For an interesting history of Konrad Zuse's contributions, see the family of sites available from http://irb.cs.tu-berlin.de/~zuse/Konrad_Zuse/en/index.html. One of these mentions that "it is undisputed that Konrad Zuse's Z3 was the first fully functional, program controlled (freely programmable) computer of the world.... The Z3 was presented on May 12, 1941, to an audience of scientists in Berlin." Instead of vacuum tubes, it used 2,400 electromechanical relays. The original Z3 was destroyed by an Allied air raid on December 21, 1943.[55] A reconstructed version was built in the early 1960s and is now on display at the Deutsche Museum in Munich. Zuse also is said to have created the first programming language, called the Plankalkül.

The American mathematician John von Neumann (1903–1957) wrote a "draft report" about the EDVAC, an early stored-program computer.[56] Perhaps because of this report, we now say that these kinds of computers use a "von Neumann architecture." The ideal von Neumann architecture separates the (task-specific) stored program from the (general-purpose) hardware circuitry, which can execute (sequentially) the instructions of any program whatsoever. (We usually call the program "software" to distinguish it from the "hardware" part of a computer. However, the distinction is blurred in most modern computers because they often have some of their programs built right into their circuitry.)

Other computers with stored programs were designed and built in the 1940s in Germany, Great Britain, and the United States. They were large, bulky machines. In Great Britain and the United States, they were mainly used for military purposes. Figure 2.21 shows one such machine.

We call computers "machines" even though today they can be made completely electrical with no moving parts whatsoever. Furthermore, when we speak of computing machines we usually mean the combination of the computer and the program it is running. Sometimes we even call just the program a machine. (As an example of this usage, I'll talk later about a "checker-playing machine" and mean a program that plays checkers.)

The commanding importance of the stored-program digital computer derives from the fact that it can be used for any purpose whatsoever – that is, of course, any computational purpose. The modern digital computer is, for all practical purposes, such a universal machine. The "all-practical-purposes" qualifier is needed because not even modern computers have the infinite storage capacity implied by Turing's infinite tape. However, they do have prodigious amounts of storage, and that makes them practically universal.

D. "Thinking" Computers

After some of the first computers were built, Turing reasoned that if they were practically universal, they should be able to do anything. In 1948 he wrote, "The importance of the universal machine is clear. We do not need to have an infinity of different machines doing different jobs. A single one will suffice. The engineering problem of producing various machines for various jobs is replaced by the office work of 'programming' the universal machine to do these jobs."[57]

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

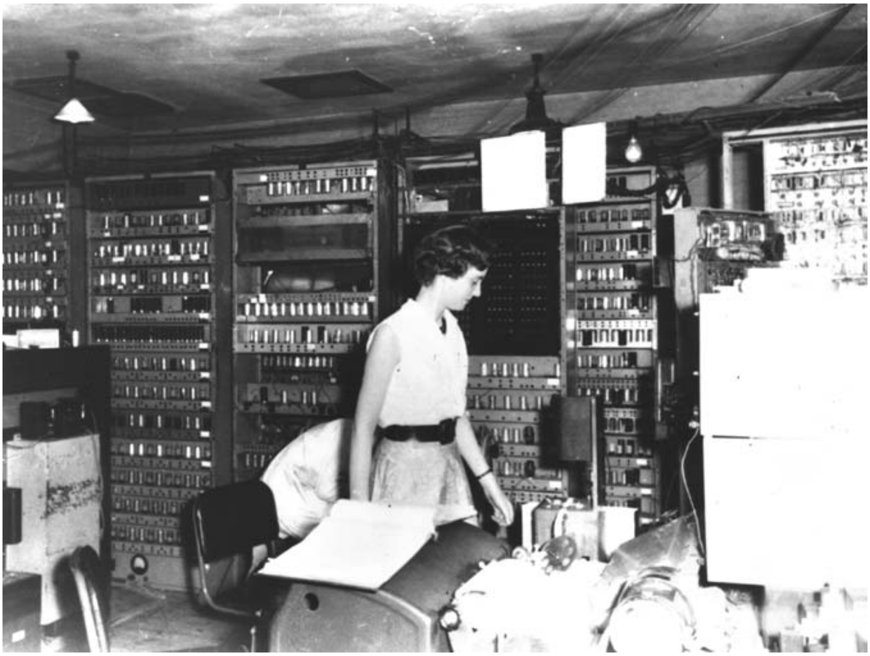

Figure 2.21. The Cambridge University EDSAC computer (circa 1949). (Copyright Computer Laboratory, University of Cambridge. Reproduced by permission.)

Among the things that Turing thought could be done by computers was mimicking human intelligence. One of Turing's biographers, Andrew Hodges, claims, "he decided the scope of the computable encompassed far more than could be captured by explicit instruction notes, and quite enough to include all that human brains did, however creative or original. Machines of sufficient complexity would have the capacity for evolving into behaviour that had never been explicitly programmed."[58]

The first modern article dealing with the possibility of mechanizing *all* of human-style intelligence was published by Turing in 1950.[59] This paper is famous for several reasons. First, Turing thought that the question "Can a machine think?" was too ambiguous. Instead, he proposed that the matter of machine intelligence be settled by what has come to be called "the Turing test."

Although there have been several reformulations (mostly simplifications) of the test, here is how Turing himself described it:

> The new form of the problem [Can machines think?] can be described in terms of a game which we call the "imitation game." It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A." The interrogator is allowed to put questions to A and B thus:
>
> C: Will X please tell me the length of his or her hair?

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

> Now suppose X is actually A, then A must answer. It is A's object in the game to try and cause C to make the wrong identification. His answer might therefore be
>
> "My hair is shingled, and the longest strands are about nine inches long."
>
> In order that tones of voice may not help the interrogator the answers should be written, or better still, typewritten. The ideal arrangement is to have a teleprinter communicating between the two rooms. Alternatively the question and answers can be repeated by an intermediary. The object of the game for the third player (B) is to help the interrogator. The best strategy for her is probably to give truthful answers. She can add such things as "I am the woman, don't listen to him!" to her answers, but it will avail nothing as the man can make similar remarks.
>
> We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?"

Most of the simplifications of Turing's test involve attempting to have a computer "in the other room" fool an interrogator into believing the computer is a human. To stimulate progress toward that goal, a prize has been established – annual prizes in fact. In the words of the Web site announcing the prizes, "In 1990 Hugh Loebner agreed with The Cambridge Center for Behavioral Studies to underwrite a contest designed to implement the Turing Test. Dr. Loebner pledged a Grand Prize of $100,000 and a Gold Medal for the first computer whose responses were indistinguishable from a human's. Such a computer can be said 'to think.' Each year an annual prize of $2000 and a bronze medal is awarded to the most human-like computer. The winner of the annual contest is the best entry relative to other entries that year, irrespective of how good it is in an absolute sense."[60]

However, several people have pointed to inadequacies of both the simplifications and even of the original test.[61] In fact, it has been alleged that even Joseph Weizenbaum's patently unintelligent 1966 conversational program, ELIZA, has fooled people into thinking it is human.[62]

For example, here is part of a transcript of a conversation with ELIZA:

ELIZA: How do you do. Please state your problem.
 > I spend too much time with computers.
ELIZA: Why do you mention computers?
 > You are one.
ELIZA: Does it please you to believe I am one?
 > Yes.
ELIZA: You seem to be quite positive.
 > I suppose I am.
ELIZA: How long have you been?
 > 31 years.
ELIZA: I'm not sure I understand you fully.
 > Join the club.
ELIZA: Please go on.
 > You don't make too much sense yourself.
ELIZA: We were discussing you – not me.

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

The first part of the conversation seems reasonable, but ELIZA bogs down in the middle because the program was expecting "I suppose I am" to be followed by some word like "happy." (There are several ELIZA simulations on the Web. One that claims to reproduce faithfully the original ELIZA program is at http://www.chayden.net/eliza/Eliza.html. Try one out!)

A second important feature of Turing's 1950 paper was his handling of arguments that people might raise against the possibility of achieving intelligent computers. I'll quote the ones Turing mentions:

(1) The Theological Objection: Thinking is a function of man's immortal soul. God has given an immortal soul to every man and woman, but not to any other animal or to machines. Hence no animal or machine can think.
(2) The 'Heads in the Sand' Objection: "The consequences of machines thinking would be too dreadful. Let us hope and believe that they cannot do so."
(3) The Mathematical Objection: There are a number of results of mathematical logic that can be used to show that there are limitations to the powers of discrete-state machines.
(4) The Argument from Consciousness: This argument is very well expressed in Professor Jefferson's Lister Oration for 1949, from which I quote:
"Not until a machine can write a sonnet or compose a concerto because of thoughts and emotions felt, and not by the chance fall of symbols, could we agree that machine equals brain – that is, not only write it but know that it had written it. No mechanism could feel (and not merely artificially signal, an easy contrivance) pleasure at its successes, grief when its valves fuse, be warmed by flattery, be made miserable by its mistakes, be charmed by sex, be angry or depressed when it cannot get what it wants."
(5) Arguments from Various Disabilities: These arguments take the form, "I grant you that you can make machines do all the things you have mentioned but you will never be able to make one to do X."
(6) Lady Lovelace's Objection: Our most detailed information of Babbage's Analytical Engine comes from a memoir by Lady Lovelace. In it she states, "The Analytical Engine has no pretensions to originate anything. It can do *whatever we know* how *to order it* to perform" (her italics).
(7) Argument from Continuity in the Nervous System: The nervous system is certainly not a discrete-state machine. A small error in the information about the size of a nervous impulse impinging on a neuron may make a large difference to the size of the outgoing impulse. It may be argued that, this being so, one cannot expect to be able to mimic the behavior of the nervous system with a discrete-state system.
(8) The Argument from Informality of Behavior: It is not possible to produce a set of rules purporting to describe what a man should do in every conceivable set of circumstances.
(9) The Argument from Extra-Sensory Perception.

In his paper, Turing nicely (in my opinion) handles all of these points, with the possible exception of the last one (because he apparently thought that extra-sensory perception was plausible). I'll leave it to you to read Turing's 1950 paper to see his counterarguments.

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

Figure 2.22. Herbert Simon (seated) and Allen Newell (standing). (Courtesy of Carnegie Mellon University Archives.)

The third important feature of Turing's 1950 paper is his suggestion about how we might go about producing programs with human-level intellectual abilities. Toward the end of his paper, he suggests, "Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain." This suggestion is really the source for the idea mentioned earlier about using an ontogenetic strategy to develop intelligent machines.

Allen Newell and Herb Simon (see Fig. 2.22) were among those who had no trouble believing that the digital computer's universality meant that it could be used to mechanize intelligence in all its manifestations – provided it had the right software. In their 1975 ACM Turing Award lecture,[63] they described a hypothesis that they had undoubtedly come to believe much earlier, the "Physical Symbol System Hypothesis." It states that "a physical symbol system has the necessary and sufficient means for intelligent action." Therefore, according to the hypothesis, appropriately programmed digital computers would be capable of intelligent action. Conversely, because humans are capable of intelligent action, they must be, according to the hypothesis, physical symbol systems. These are very strong claims that continue to be debated.

Both the imagined Turing machine and the very real digital computer are symbol systems in the sense Newell and Simon meant the phrase. How can a Turing machine, which uses a tape with 0's and 1's printed on it, be a "symbol system"? Well, the 0's and 1's printed on the tape can be thought of as symbols standing for their associated

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

numbers. Other symbols, such as "A" and "M," can be encoded as sequences of primitive symbols, such as 0's and 1's. Words can be encoded as sequences of letters, and so on. The fact that one commonly thinks of a digital computer as a machine operating on 0's and 1's need not prevent us from thinking of it also as operating on more complex symbols. After all, we are all used to using computers to do "word processing" and to send e-mail.

Newell and Simon admitted that their hypothesis could indeed be false: "Intelligent behavior is not so easy to produce that any system will exhibit it willy-nilly. Indeed, there are people whose analyses lead them to conclude either on philosophical or on scientific grounds that the hypothesis is false. Scientifically, one can attack or defend it only by bringing forth empirical evidence about the natural world." They conclude the following:

> The symbol system hypothesis implies that the symbolic behavior of man arises because he has the characteristics of a physical symbol system. Hence, the results of efforts to model human behavior with symbol systems become an important part of the evidence for the hypothesis, and research in artificial intelligence goes on in close collaboration with research in information processing psychology, as it is usually called.

Although the hypothesis was not formally described until it appeared in the 1976 article, it was certainly implicit in what Turing and other researchers believed in the 1950s. After Allen Newell's death, Herb Simon wrote, "From the very beginning something like the physical symbol system hypothesis was embedded in the research."[64]

Inspired by the clues we have mentioned and armed with the general-purpose digital computer, researchers began, during the 1950s, to explore various paths toward mechanizing intelligence. With a firm belief in the symbol system hypothesis, some people began programming computers to attempt to get them to perform some of the intellectual tasks that humans could perform. Around the same time, other researchers began exploring approaches that did not depend explicitly on symbol processing. They took their inspiration mainly from the work of McCulloch and Pitts on networks of neuron-like units and from statistical approaches to decision making. A split between symbol-processing methods and what has come to be called "brain-style" and "nonsymbolic" methods still survives today.

Notes

1. Aristotle, *Prior Analytics, Book I*, written circa 350 BCE, translated by A. J. Jenkinson, Web addition published by eBooks@Adelaide, available online at http://etext.library.adelaide.edu.au/a/aristotle/a8pra/. [10]
2. Medieval students of logic gave names to the different syllogisms they studied. They used the mnemonic *Barbara* for this one because each of the three statements begins with "All," whose first letter is "A." The vowels in "Barbara" are three"a"s. [10]
3. From Martin Davis, *The Universal Computer: The Road from Leibniz to Turing*, New York: W.W. Norton & Co., 2000. For an excerpt from the paperback version containing this quotation, see http://www.wwnorton.com/catalog/fall01/032229EXCERPT.htm. [11]

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

4. Quotation from William Aspray (ed.), *Computing Before Computers*, Chapter 3, "Logic Machines," pp. 107–8, Ames, Iowa: Iowa State Press, 1990. (Also available from http://ed-thelen.org/comp-hist/CBC.html.) [12]
5. Robert Harley, "The Stanhope Demonstrator," *Mind*, Vol. IV, pp. 192–210, 1879. [13]
6. George Boole, *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*, Dover Publications, 1854. [13]
7. See D. McHale, *George Boole: His Life and Work*, Dublin, 1985. This excerpt was taken from http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Boole.html. [14]
8. See, for example, Gerard O'Regan, *A Brief History of Computing*, p. 17, London: Springer-Verlag, 2008. [14]
9. I follow the pictorial version used in the online Stanford Encyclopedia of Philosophy (http://plato.stanford.edu/entries/frege/), which states that ". . . we are modifying Frege's notation a bit so as to simplify the presentation; we shall not use the special typeface (Gothic) that Frege used for variables in general statements, or observe some of the special conventions that he adopted. . . . " [14]
10. Warren S. McCulloch and Walter Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133, Chicago: University of Chicago Press, 1943. (See Marvin Minsky, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice-Hall, 1967, for a very readable treatment of the computational aspects of "McCulloch–Pitts neurons.") [17]
11. Donald O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, New York: John Wiley, Inc., 1949. [17]
12. For more about Hebb, see http://www.cpa.ca/Psynopsis/special_eng.html. [18]
13. For a summary of the lives and work of both men, see a Web page entitled "Wilhelm Wundt and William James" by Dr. C. George Boeree at http://www.ship.edu/~cgboeree/wundtjames.html. [19]
14. M. Minsky (ed.), "Introduction," *Semantic Information Processing*, p. 2, Cambridge, MA: MIT Press, 1968. [20]
15. Russell A. Kirsch, "Experiments with a Computer Learning Routine," Computer Seminar Notes, July 30, 1954. Available online at http://www.nist.gov/msidlibrary/doc/kirsch_1954_artificial.pdf. [20]
16. B. F. Skinner, *Verbal Behavior*, Engelwood Cliffs, NJ: Prentice Hall, 1957. [20]
17. Noam Chomsky, "A Review of B. F. Skinner's *Verbal Behavior*," in Leon A. Jakobovits and Murray S. Miron (eds.), *Readings in the Psychology of Language*, Engelwood Cliffs, NJ: Prentice-Hall, 1967. Available online at http://www.chomsky.info/articles/1967—.htm. [20]
18. See, for example, N. Chomsky, *Aspects of the Theory of Syntax*, Cambridge: MIT Press, 1965. [21]
19. George A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *The Psychological Review*, Vol. 63, pp. 81–97, 1956. [21]
20. *IRE Transactions on Information Theory*, Vol IT-2, 1956. [21]
21. For a copy of his paper, see http://www.chomsky.info/articles/195609–.pdf. [21]
22. George A. Miller, "A Very Personal History," MIT Center for Cognitive Science Occasional Paper No. 1, 1979. [21]
23. George A. Miller, E. Galanter, and K. H. Pribram, *Plans and the Structure of Behavior*, New York: Holt, Rinehart & Winston, 1960. [21]
24. For a thorough history of cognitive science, see Margaret A. Boden, *Mind As Machine: A History of Cognitive Science*, vols. 1 and 2, Oxford: Clarendon Press, 2006. For an earlier,

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

one-volume treatment, see Howard E. Gardner, *The Mind's New Science: A History of the Cognitive Revolution*, New York: Basic Books, 1985. [22]

25. An English translation appeared later: N.A. Barricelli, "Symbiogenetic Evolution Processes Realized by Artificial Methods," *Methodos*, Vol. 9, Nos. 35–36, pp. 143–182, 1957. For a summary of Barricelli's experiments, see David B. Fogel, "Nils Barricelli – Artificial Life, Coevolution, Self-Adaptation," *IEEE Computational Intelligence Magazine*, Vol. 1, No. 1, pp. 41–45, February 2006. [22]
26. R. M. Friedberg, "A Learning Machine: Part I," *IBM Journal of Research and Development*, Vol. 2, No. 1, pp. 2–13, 1958, and R. M. Friedberg, B. Dunham, and J. H. North, "A Learning Machine: Part II," *IBM Journal of Research and Development*, Vol. 3, No. 3, pp. 282–287, 1959. The papers are available (for a fee) at http://www.research.ibm.com/journal/rd/021/ibmrd0201B.pdf and http://www.research.ibm.com/journal/rd/033/ibmrd0303H.pdf. [22]
27. Marvin L. Minsky, "Steps Toward Artificial Intelligence," *Proceedings of the Institute of Radio Engineers*, Vol. 49, pp. 8–30, 1961. Paper available at http://web.media.mit.edu/~minsky/papers/steps.html. [23]
28. Lawrence J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, New York: Wiley, 1966. [23]
29. Woodrow W. Bledsoe, "The Evolutionary Method in Hill Climbing: Convergence Rates," Technical Report, Panoramic Research, Inc., Palo Alto, CA, 1962.; Hans J. Bremermann, "Optimization through Evolution and Recombination," M. C. Yovits, G. T. Jacobi, and G. D. Goldstein (eds.), *Self-Organizing Systems*, pp. 93–106, Washington, DC: Spartan Books, 1962. [23]
30. Jürgen Schmidhuber, "2006: Celebrating 75 Years of AI – History and Outlook: The Next 25 Years," in Max Lungarella et al. (eds.), *50 Years of Artificial Intelligence: Essays Dedicated to the 50th Anniversary of Artificial Intelligence*, Berlin: Springer-Verlag, 2007. Schmiduber cites Ingo Rechenberg, "Evolutionsstrategie – Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution," Ph.D. dissertation, 1971 (reprinted by Frommann-Holzboog Verlag, Stuttgart, 1973). [23]
31. See http://www.aaai.org/AITopics/html/genalg.html. [23]
32. John H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan Press, 1975. Second edition, MIT Press, 1992. [23]
33. W. Grey Walter, "An Imitation of Life," *Scientific American*, pp. 42–45, May 1950. See also W. Grey Walter, *The Living Brain*, London: Gerald Duckworth & Co. Ltd., 1953. [24]
34. B. Inhelder and J. Piaget, *The Growth of Logical Thinking from Childhood to Adolescence*, New York: Basic Books, 1958. For a summary of these stages, see the following Web pages: http://www.childdevelopmentinfo.com/development/piaget.shtml and http://www.ship.edu/~cgboeree/piaget.html. [24]
35. *Proceedings of the Bionics Symposium: Living Prototypes – the Key to new Technology,* Technical Report 60-600, Wright Air Development Division, Dayton, Ohio, 1960. [25]
36. *Proceedings of the Third Bionics Symposium*, Aerospace Medical Division, Air Force Systems Command, United States Air Force, Wright-Patterson AFB, Ohio, 1963. [25]
37. http://www.mlahanas.de/Greeks/Ctesibius1.htm. [27]
38. http://www.asc-cybernetics.org/foundations/timeline.htm. [27]
39. From http://www.nickgreen.pwp.blueyonder.co.uk/control.htm. [27]
40. For a history of cybernetics, see a Web page of the American Society for Cybernetics at http://www.asc-cybernetics.org/foundations/history.htm. [28]
41. From http://pespmc1.vub.ac.be/ASC/SELF-ORGANI.html. [28]

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

42. W. Ross Ashby, “Principles of the Self-Organizing Dynamic System,” *Journal of General Psychology*, Vol. 37, pp. 125–128, 1947. See also the Web pages at http://en.wikipedia.org/wiki/Self_organization. [29]
43. Bayes wrote an essay that is said to have contained a version of the rule. Later, the Marquis de Laplace (1749–1827) generalized (some say independently) what Bayes had done. For a version of Bayes’s essay (posthumously written up by Richard Price), see http://www.stat.ucla.edu/history/essay.pdf. [29]
44. David Van Meter and David Middleton, “Modern Statistical Approaches to Reception in Communication Theory,” Symposium on Information Theory, *IRE Transactions on Information Theory*, PGIT-4, pp. 119–145, September 1954. [31]
45. William Aspray (ed.), *Computing Before Computers*, Ames, Iowa: Iowa State University Press, 1990. Available online at http://ed-thelen.org/comp-hist/CBC.html. [31]
46. *Ibid*, Chapter 1. [31]
47. *Ibid*. [32]
48. *Ibid*, Chapter 2. [33]
49. Alonzo Church, “An Unsolvable Problem of Elementary Number Theory,” *American Journal of Mathematics*, Vol. 58, pp. 345–363, 1936. [33]
50. Alan M. Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society*, Series 2, Vol. 42, pp. 230–265, 1936–1937. [33]
51. For more information about Turing, his life and works, see the Web pages maintained by the Turing biographer, Andrew Hodges, at http://www.turing.org.uk/turing/. [33]
52. The quotation is from Alan M. Turing, “Lecture to the London Mathematical Society,” p. 112, typescript in King’s College, Cambridge, published in Alan M. Turing’s *ACE Report of 1946 and Other Papers* (edited by B. E. Carpenter and R. W. Doran, Cambridge, MA: MIT Press, 1986), and in Volume 3 of *The Collected Works of A. M. Turing* (edited D. C. Ince, Amsterdam: North-Holland 1992). [35]
53. For a biographical sketch, see http://www.research.att.com/~njas/doc/shannonbio.html. [35]
54. In his book *The Mind’s New Science*, Howard Gardner called this thesis “possibly the most important, and also the most famous, master’s thesis of the century.” [35]
55. Various sources give different dates for the air raid, but a letter in the possession of Zuse’s son, Horst Zuse, gives the 1943 date (according to an e-mail sent me on February 10, 2009, by Wolfgang Bibel, who has communicated with Horst Zuse). [36]
56. A copy of the report, plus introductory commentary, can be found from http://qss.stanford.edu/~godfrey/. [36]
57. Alan M. Turing, “Intelligent Machinery,” National Physical Laboratory Report, 1948. Reprinted in B. Meltzer and D. Michie (eds), *Machine Intelligence 5*, Edinburgh: Edinburgh University Press, 1969. A facsimile of the report is available online at http://www.AlanTuring.net/intelligent_machinery. [36]
58. Andrew Hodges, *Turing*, London: Phoenix, 1997. [37]
59. Alan M. Turing, “Computing Machinery and Intelligence,” *Mind*, Vol. LIX, No. 236, pp. 433–460, October 1950. (Available at http://www.abelard.org/turpap/turpap.htm.) [37]
60. See the “Home Page of the Loebner Prize in Artificial Intelligence” at http://www.loebner.net/Prizef/loebner-prize.html. [38]
61. For discussion, see the Wikipedia article at http://en.wikipedia.org/wiki/Turing_test. [38]
62. Joseph Weizenbaum, “ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine,” *Communications of the ACM*,

https://doi.org/10.1017/CBO9780511819346.003 Published online by Cambridge University Press

Vol. 9, No. 1, pp. 36–35, January 1966. Available online at http://i5.nyu.edu/∼mm64/x52.9265/january1966.html. [38]

63. Allen Newell and Herbert A. Simon, "Computer Science as Empirical Inquiry: Symbols and Search," *Communications of the ACM*, Vol. 19, No. 3, pp. 113–126, March 1976. [40]

64. National Academy of Sciences, *Biographical Memoirs*, Vol. 71, 1997. Available online at http://www.nap.edu/catalog.php?record_id=5737. [41]