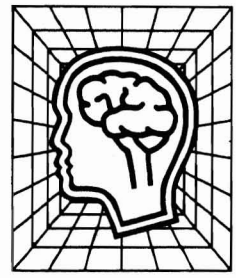
2

The Brain and the Computer



The human brain is the most highly organized and complex structure in the known universe. What do we really know about this remarkable organ and where does this knowledge come from?

Our understanding of the human brain is based on:

1. Physiological and psychological investigations, going back at least 2500 years[6] to the work of Hippocrates, that attempt to catalog and relate brain structure and function by experiment and direct observation
2. Analogy to the mechanical devices built by man that attempt to duplicate some of the brain's functional abilities

We will review some of the anatomical knowledge about the brain's architecture, but there is little hope that the structures we can currently observe and describe will shed much light on how the brain really functions. In a device as complex as the brain, function is too deeply encoded in structure to be deciphered without already knowing the relationships for which we are searching. We can trace some of the sensory and motor pathways for a short distance into the brain, but once we pass beyond the point of direct sensor signal transmission, conditioning, and reflex behavior, we have

[6]Knowledge of the neurological symptoms resulting from specific brain injuries existed as early as 3000 B.C. For example, the Edward Smith Papyrus, a surgical treatise, describes the location of certain sensory and motor control areas in the brain.

little understanding of what the brain is actually doing.

At present, our best hope for understanding the brain and the nature of human intelligence appears to be through analogy with the computer and the associated mathematical theory of computation. This may be a false hope, both with respect to understanding and to man's attempts to build an intelligent device in his own image. Historically, attempts have been made to explain the brain's behavior in terms of the most advanced artifacts of the time: in terms of clockwork mechanisms, telephone switchboard analogies, and now the digital computer.

One of our main goals in this chapter is to address the question of whether there are essential differences between the brain and the computer that will prevent machine intelligence from reaching human levels of achievement. In particular, we examine the ultimate capacity of the computer as an intelligence engine:

(a) To what extent is the computer an adequate model for explaining the functioning and competence of the brain?
(b) Are there problems that cannot be solved (in practice or in theory) by a logical device?
(c) Is there a limit to the complexity of a physical device beyond which unreliability renders it successively less (rather than more) competent?

THE HUMAN BRAIN

The human brain is constructed out of more than 10 billion individual components (nerve cells). Can we really hope to understand how something so complex operates, or even determine what it is doing or trying to accomplish? Our current view, that the brain controls the body and is the seat of consciousness, and our understanding of the nature of intelligence and intelligent behavior, is still developing.

In this section we first discuss the evolution of the brain and present a model of its organization based on an evolutionary perspective, the so-called triune brain of MacLean. Next, we describe the architecture of the brain and present two functionally oriented models, one due to Luria, and the second, with more of a philosophical flavor, due to Penfield.

Evolution of the Brain

How did the brain evolve? Is there a continuous spectrum of elaboration reaching from the simplest organisms to man, or is there a sequence of distinct "inventions" that sharply partitions the competence of the organisms with brains incorporating these inventions?

Living organisms have evolved two distinct strategies for obtaining the food and energy necessary to sustain life. Plants are stationary factories that exploit the largely renewable nonliving resources in their environment. Animals eat other living things and must be capable of both finding and catching their prey—i.e., of perception and motion. The physiological correlates of purposive movement through the environment are sensors, muscles, and an effective apparatus for interpretation, coordination, and control.

The essential invention that allowed higher-level animal life to evolve was the

nerve cell (Appendix 2-1), and indeed, one of the most important distinctions between animal and plant life (once we pass beyond the most primitive organisms) is that animals possess nervous tissue and plants do not. The nerve cell provides a way of rapidly transmitting sensed information and muscular control commands using a unique combination of electrical and chemical signals, while in plant life, coordination of activity is accomplished exclusively by much slower chemical messages. In addition to a speed advantage, nervous tissue possesses an unusual degree of "plasticity" (modification of function due to environmental influences) that seems to provide the basis for learning.

The first simple animals, like plants, were passive organisms, either stationary or drifters—moved mostly by wind or tide. It is believed that one of man's most distant ancestors was a miniscule wormlike creature that floated in the surface layers of the warm Cambrian seas some 500 million years ago, and that a strip of light-sensing cells and associated neurons developed on its dorsal (back) surface to improve its ability to properly orient and position itself relative to the surface illumination. This strip of nerve cells, by creasing and folding inward (invaginating), first formed a tubular nerve cord and eventually evolved into the spinal cord that distinguishes the vertebrates, including the higher forms of animal life, and ultimately man. (See Table 2-1, Fig. 2-1

TABLE 2-1 ■ The Evolution of Animal Life

Era	Period	Epoch	Years Before Present (millions)	Life Forms
Cenozoic	Quaternary	Holocene Pleistocene	3	Modern man Early man
	Tertiary	Pliocene Miocene Oligocene Eocene Paleocene	70	Large carnivores Grazing mammals Large mammals Modern mammals Early mammals, modern birds
Mesozoic	Cretaceous		130	Climax of reptiles, conifers, first flowering plants
	Jurassic		165	First true mammals, first birds
	Triassic		200	First dinosaurs, amphibians
Paleozoic	Permian		230	Abundant insect life
	Pennsylvanian		300	First reptiles
	Mississippian		320	Sharks
	Devonian		360	First amphibians
	Silurian		400	First land plants
	Ordovician		480	First fishes
	Cambrian		550	Abundant marine life
Precambrian			600	Very primitive organisms (Few fossils found)

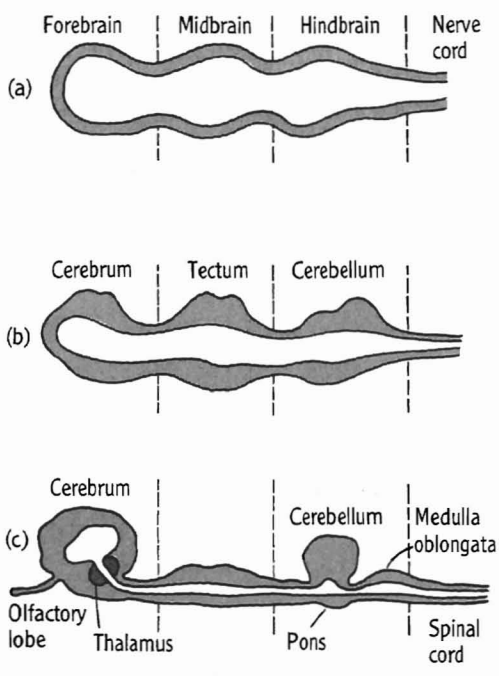


FIGURE 2-1
Evolution of the Vertebrate Brain.

(a) Diagrammatic depiction of the three primary swellings of the neural tube as it is believed to exist in aboriginal chordates, and as it appears in embryonic development of the human brain. (b) Evolutionary developments believed to have occurred in the roof of the primitive neural tube. (After C. M. U. Smith. *The Brain, Towards an Understanding*. Capricorn Books, New York, 1972.) (c) Elaboration of the neural tube in embryonic development of the human brain.

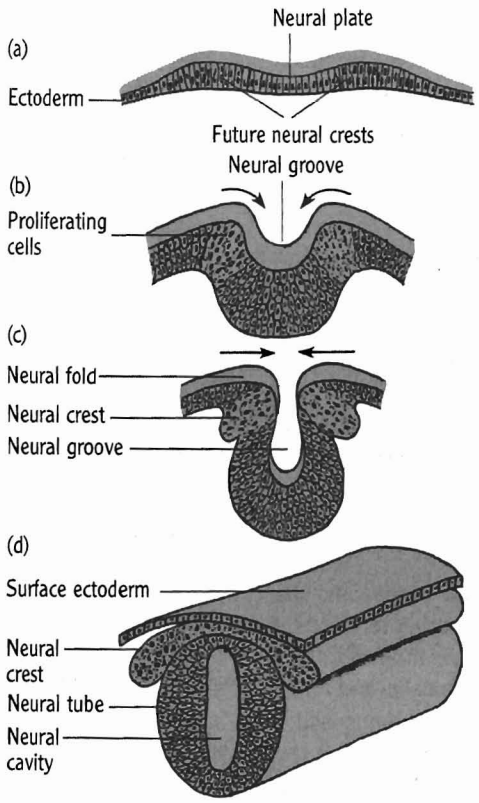
and 2-2, and Box 2-1). In the course of evolutionary development, sense organs tended to develop on the forward (anterior) end of the organisms, for that is the end that first penetrates new environments. Nerve centers concerned with analysis of data from these sensory organs also moved forward to minimize the overall data transmission requirements and time delays, a process called "neurobiotaxis."

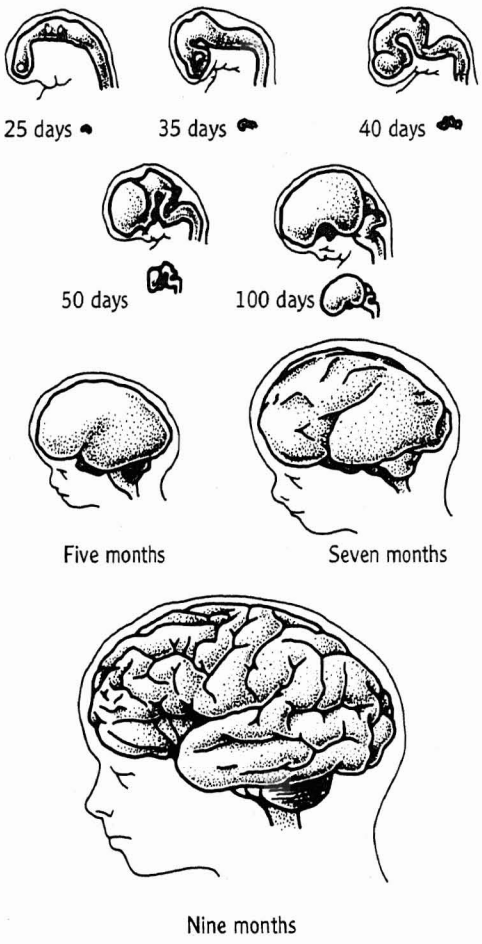
It appears that the aboriginal vertebrate brain (the somewhat enlarged anterior end of the spinal cord) underwent a series of three evolutionary expansions to permit the development of the three main distance receptors (See Fig. 2-1): the hindbrain for vibration and sound, the midbrain for vision, and the forebrain for olfaction (smell). In the higher vertebrates, and especially man with his elaborated cortex, the sensory interpretation functions migrated from the lower centers where they originally evolved, and now mainly reside in the cortex itself. Nevertheless, in the growth of the individual, the vastly more complex modern chordate brain still develops from these three bulges in the embryonic neural tube (Fig. 2-I and 2-2). The hindbrain gives rise to the cerebellum, the main center for muscular coordination; the midbrain enlarges into the optic tectum, which still serves as the main visual center in birds and fish; and the forebrain, which grows into the large multifunction cerebrum in man, is an inconspicuous swelling in many lower vertebrates that is employed to analyze the inputs from their olfactory organs (Color Plate 1). It should be noted that olfaction is the dominant sense in most mammals. Food selection, hunting, socializing, mating, and navigation can all be effectively based on a keen sense of smell. Almost alone among mammals, vision dominates smell in the primates. This is undoubtedly due to the fact that the primates evolved in the trees where three-dimensional vision is critical to survival, and scents quickly fade.

FIGURE 2-2

Part I: Embryonic Development of the Nervous System.

(a) Ectodermal cells form a neural plate in the midline, and proliferate to form a multicellular layer. (b) As cells at each side of the neural plate proliferate, the sides are elevated (arrows) to form neural folds enclosing a groove. (c) The neural groove deepens and the neural folds come together in the midline (arrows), fuse, and form the neural tube. (d) The neural tube forms the primitive central nervous system. The overlying neural crest will form the peripheral nervous system and related cells. (From E. L. Weinreb. *Anatomy and Physiology*. Addison-Wesley, Reading, Mass., 1984, p.158, with permission.)





Part II: Embryonic Development of the Human Brain.

The three main parts of the brain (the forebrain, the midbrain and the hindbrain) originate as prominent swellings at the head end of the early neural tube. The cerebral hemispheres eventually overgrow the midbrain and the hind brain and also partly obscure the cerebellum. The characteristic convolutions and invaginations of the brain's surface do not begin to appear until about the middle of pregnancy. (From *Scientific American*, 1979. Reprinted by permission.)

BOX 2-1 Animal Evolution

The human brain developed in the context of animal evolution. The story of this evolution can be told in terms of a series of "inventions" involving not only sensory and integrative systems (based on the original invention of the nerve cell), but also inventions with respect to:

1. Heredity and reproduction—e.g., DNA, sex
2. Skeletal, effector, and locomotion systems—e.g., bones, muscles, skin, hair, spinal column and vertebrae, legs, arms, fingers, opposing thumb
3. Energy acquisition and utilization via internal transport systems—e.g., ATP, lungs, blood, digestive enzymes, alimentary tract
4. Systems for internal regulation and body maintenance—e.g., the immune system, control of temperature, breathing, heart rate and blood flow, thirst, hunger, emotions

While no one invention stands by itself, a system of classification based on some of the more obvious and easily observable inventions has been devised to distinguish the various life forms and their evolutionary progression. The classification of man is shown in the Table 2-2. Further discussion can be found in Wasserman [Wasserman 73].

TABLE 2-2 ■ Classification of Man

ORGANISM: Man
KINGDOM: Animal (Other kingdoms are plant, Protista, Monera.)
PHYLUM: Chordata (Distinguished by a backbone or notochord, a longitudinal stiffening rod which lies between the central nervous system and the alimentary canal; a hollow, dorsal nerve cord; and embryonic gill slits. The chordates include over 70,000 species distributed over four subphyla. Other major phyla include the Arthropoda, Mollusca, and Echinodermata.
SUBPHYLUM: Vertebrata (The embryonic notochord is replaced by a backbone of vertebrae as the central axis of the endoskeleton.)
CLASS: Mammalia (Warm-blooded; air-breathing; milk-producing; four-chambered heart; possesses hair; young born alive. Other classes include fish, amphibians, reptiles, and birds.)
ORDER: Primates (Enlarged cranium with eyes located on front of head; stands erect; thumbs opposing the fingers; fingers have nails instead of claws. Other orders include rodents and carnivores.)
FAMILY: Hominidae (Large cerebral hemispheres overhanging the cerebellum and medulla. Apes belong to the family Pongidae which consists of the gorilla, chimpanzee, orangutan, and gibbon. There are two separate families of monkeys that also include the baboons.)
SPECIES: *Homo sapiens* (Man is the only living species of the family Hominidae.)

One of the more interesting accounts of the present structure of the human brain, based on evolutionary development, is due to MacLean [MacLean 73]. He hypothesizes that the brain consists of three interconnected biological computers (the "triune brain," Fig. 2-3), each with its own type of intelligence, subjectivity, sense of time and space, memory, motor, and other functions. Each of these three brains (known to be distinct anatomically, chemically, and functionally) corresponds to a separate evolutionary step. The combination of spinal cord, hindbrain, and midbrain (collectively called the "neural chassis") contains the neural machinery

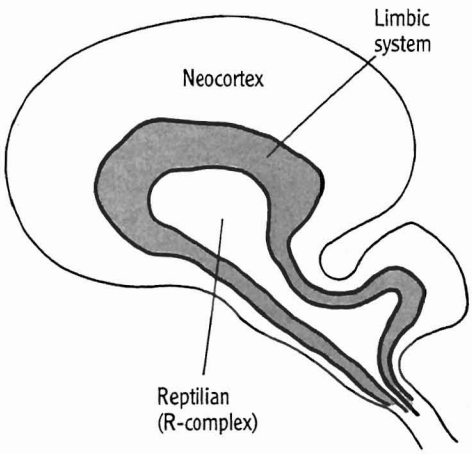


FIGURE 2-3 The Triune Brain.

General schematic of the three major components of the triune brain. (After P. D. MacLean. *A Triune Concept of Brain and Behavior.* University of Toronto Press, Toronto, 1973.)

necessary for reproduction and self-preservation, including control of temperature, muscle tone, sleep rhythm, blood circulation, and respiratory functions. In a fish or amphibian, this is almost all the brain there is; however, more highly evolved organisms are animated by "superior" brain structures, and are reduced to a vegetative state when these higher brain structures are rendered inoperative. MacLean distinguishes three separate drivers of the neural chassis:

1. The reptilian or R-complex, which probably evolved 200 to 300 million years ago, consists of the structures immediately surrounding the midbrain (*corpus striatum, globus pallidus*). We share this complex with other mammals and reptiles. It plays a major role in aggressive behavior, territoriality, ritual, and the establishment of social hierarchies. It is surprising how much of modern human conduct can be ascribed to these primitive behavior patterns.
2. The limbic system (*thalamus, hypothalamus, hippocampus, amygdala, pituitary*), which evolved more than 150 million years ago, is located on top of the R-complex. We share the limbic system with other mammals, but some of its structures are not possessed by reptiles. The limbic system appears to be the site of emotional response (fear, love, hate, pleasure and especially sexual pleasure, pain, altruism, sentiment) and is a major center for memory storage and recall. The oldest part of the limbic system, the olfactory cortex, which originally evolved to analyze scents and smells, still serves in this capacity. The role of smell in sexual behavior, and its involvement in memory, is not accidental.
3. The neocortex, sitting like a cap on the rest of the brain, evolved in the last 50 million years, but the rate of its evolutionary growth increased dramatically in the last few million years in the primates and especially in man.

MacLean has based his theory on years of careful study of the behavior of animals, ranging from lizards to squirrel monkeys, in which he determined which parts of the brain control what types of

behavior. Further, his theory of evolution by addition and preservation of preexisting structure is also justified, in part, by the argument that it is very difficult to evolve by randomly altering a complex system—any such change is likely to be lethal. However, fundamental change can be accomplished by the addition of new systems to the old ones.

Architecture of the Brain

The human brain (See Box 2-2, Table 2-3, and Color Plate 1) is dominated by a massive cortex, which is bilaterally symmetrical in structure. Each cortical hemisphere is composed of four major regions or lobes. These are named the *frontal, parietal, temporal,* and *occipital* lobes. While it is clear that these lobes do not act as independent functional units, (most higher level functions are known to be distributed across more than one region of the cortex), it is still the case that many human attributes and functions appear to be strongly associated with a single lobe.

The frontal lobes appear to be associated with initiative, anticipation, caution, and the general regulation and planning of action; the temporal lobes with the integration of perceptual information, especially speech and vision; the parietal lobes with symbolic processes (reading, writing, arithmetic), spatial perception, and motor control; and the occipital lobes with vision, the dominant sense in humans and other primates.

Man has convolutions in his cerebral

TABLE 2-3 ■ Physical Attributes of the Human Brain

Attribute	The Brain
Types of processing elements	Neuron: up to 100 distinct classes; functional differences not known
Number of elements	10^{10} to 10^{12} neurons
Size/volume	Brain volume (man) = 1500 cc Neuron (cell) body diameter = 0.004 in. Axon length: up to a few feet
Weight	3.3 lb
Power	10 watts
Transmission and switching speed	Transmission speed: function of axon diameter and insulation, ranges from 30 to 360 ft/sec Maximum switching speed 0.5×10^{-3} sec
Interconnection complexity per computing element	Up to 200,000 connections (for Purkinje cell)
Reliability	Component reliability: low, neurons dying continuously System reliability: high, design life 70+ yr
Information coding	Digital: frequency modulation

BOX 2-2 Structural Organization of the Mammalian Neocortex

The neocortex is remarkably similar in the brains of all mammals, including man. The same cell types are found and the same stratified structure with six parallel layers* is observed.

Numbered from one to six from the surface inward, layer 1 contains mostly fibers from neurons in other layers; layers 2, 3, and the upper portion of layer 5 deal with internal processing; layer 4 is largely involved in receiving sensory information; and the lower portion of layer 5 and all of layer 6 are concerned with muscle control. The thickness of the neocortex varies somewhat in different brain regions ranging between 50 to 100 cells in depth.

The number of cells lying beneath a fixed-size patch of surface area is essentially constant for all areas of the neocortex: 140,000 neurons per square millimeter of surface with the exception of the visual area where primates have 2.5 times as many cells as in other areas. The human neocortex has a surface area of about 2200 cm^2 and is estimated to contain 30 billion neurons. The corresponding numbers for the chimp and gorilla are 500 cm^2 and 7.5 billion cortical neurons; the cat has 4 to 5 cm^2 of cortex containing 65 million neurons. The average thickness of the neocortex increases by a factor of three in the evolutionary progression from rat to man, reflecting an increase in the amount of "wiring" needed to interconnect the larger number of neurons; however, the density of synapses seems to have remained unchanged.

Thus, the human brain is not visibly distinguished in either gross structural formation, cell type, cell distribution, cell density, or density of synapses, as we ascend the evolutionary scale from the lower mammals. The major visible evolutionary change is the continuous quantitative increase in neocortex surface area, thickness, total number of neurons, and the total number of connections between neurons. From fish to man, the brain assumes an increasingly greater fraction of body weight. In mammals, the neocortex size (or equivalent surface area) shows a similar evolutionary increase relative to the total brain size [Changeux 85, Smith 72].

*From a functional standpoint, the vertical organization of the cortex is as important as the layered horizontal organization: the entire neocortex seems to consist of a mosaic of overlapping functional columns. The vertical organization is described in Chapter 8 for the visual system.

cortex that are new from the point of view of evolution, and not committed to motor or sensory functions. These areas, which are "programmed" to function after birth, are primarily in the prefrontal and temporal lobes. During childhood, some of this uncommitted area on one side or the other (but usually the left side) of the temporal lobes will be programmed for speech. The remaining area, called the interpretation cortex, is apparently reserved for the interpretation of present events in the light of past experience.

A theory that attempts to characterize the functional organization of the brain is due to Luria [Luria 73]. He describes three main functional units. The first unit, centered mainly in the upper brain stem (especially the reticular formation) and in the limbic region, is concerned with the maintenance and regulation of the general "tone" or level of activity in the brain, and more generally, with consciousness and emotion.

The second unit is concerned with modeling the relation of the organism to the external world, and thus with the interpretation and storage of sensory

information. This second unit is composed of independent subsystems for each of the different sensory modalities (e.g., visual, auditory, cutaneous, and kinesthetic senses). However, each of these subsystems is organized along similar architectural lines: each sensory modality has a primary reception area that organizes information received directly from the sensory organs. A secondary region, also specific to each sensory modality, appears to interpret the primary sensory output in the light of stored knowledge and past experience, and is responsible for the symbolic encoding of the sensory signals. Finally, a tertiary area, shared among the different senses, integrates symbolic information from the different sensory modalities in creating a composite model of the world. Luria asserts that this tertiary area is a unique human brain structure that converts concrete perception into abstract verbal thinking employing some of the same machinery associated with the speech function.

Luria's third unit, centered largely in the frontal lobes, is concerned with the formation of intentions, the creation of plans, and the monitoring of performance. The third unit controls the actions and thus the motor systems of the organism. Again, Luria asserts that the frontal (prefrontal) lobes, much more highly developed in man than in any other animal (occupying up to one quarter of the total mass of the human brain), are organized to employ symbols and speech processes in their functioning.

Rather than continuing to catalog our admittedly limited knowledge of the relationships between brain structure and function, in the remainder of this subsection we will give a brief speculative account of what is known about the highest and most fascinating brain functions: mind, consciousness, personality, pleasure and pain, learning and memory, and reasoning. Perception is discussed extensively in a later chapter.

Mind, Personality, Consciousness, and the Soul. Each human brain appears to house a single individual, although there are rare pathological cases of multiple personalities alternately manifesting their presence in a single body. How do 10 billion nerve cells interact to produce a single consciousness? Where is the site of the "I," conscious awareness, or even the mind or soul should one or the other exist independent of the physical structures of the brain?

In a view contrary to that of the "triune brain" as hypothesized by MacLean, and also distinct from that of Luria, Wilder Penfield [Penfield 78] believes that the brain is a tightly integrated whole, and that conscious awareness resides not in the new brain (neocortex) but rather in the old (the brain stem). In Penfield's theory, the brain consists of two major systems, (1) the mechanisms associated with the existence and maintenance of conscious awareness, the mind, and (2) the mechanisms involved in sensorymotor coordination, called the central integrating system.

Penfield, one of the world's foremost neurologists/neurosurgeons at the time of his death in 1976, formulated his views after a lifetime of studying how the brain functions and malfunctions, especially in the presence of epilepsy. He observed that epileptic fits, abnormal and uncontrolled electrical discharges in the brain that disable the affected areas, generally limit

themselves to one functional system. One such type of epileptic fit, called a *petit mal*,[7] converts the individual into an automaton. The patient becomes unconscious, but may wander about in an aimless manner or he may continue to carry out whatever task he had started before the attack, following a stereotyped pattern of behavior. He can make few, if any, decisions for which there has been no precedent, and makes no record of a stream of consciousness—he will have complete amnesia for the period of epileptic discharge. The regions of the brain affected by petit mal are the prefrontal and temporal lobes and the gray matter in the higher brain stem. When an epileptic discharge occurs in the cerebral cortex in any of the sensory or motor areas (e.g., in the parietal or occipital lobes), and spreads to the higher brain stem, the result is always a major convulsive attack (*grand mal*), never an attack of "automatism."

We note that both the central integrating system (essentially a computer) and the mechanisms responsible for mind (consciousness, awareness) have primary, but distinct, centers in the gray matter of the higher brain stem (diencephalon) where they engage in a close functional relationship. With the exception of pain and possibly smell sensations, which make no detour to the cerebral cortex, all sensory signals come first to the higher brain stem, and then continue on to an appropriate region of the cerebral cortex; from there, they return to specific areas of the diencephalon. Thus, according to Penfield, the cerebral cortex, instead of being the highest level of integration, is an elaboration layer, partitioned into distinct functional areas.

The indispensable machinery that supports consciousness lies outside of the cerebral cortex: removal of large portions of the cerebral cortex does not cause loss of consciousness, but injury or interference with function in the higher brain stem, even in small areas, abolishes consciousness completely.

In summary, Penfield views the sensory interpretation and motor control areas of the cerebrum as a "computer" that operates in the service of the "mind." The structures that support the highest function of the brain, conscious awareness, are thought to be located primarily in the higher brain stem and in the "uncommitted areas" of the cerebrum (especially the prefrontal and temporal lobes). Even if Penfield is correct, we still understand very little about the nature of conscious awareness, nor do we have any definitive way of answering questions such as: At what point in evolutionary development did conscious awareness first arise, and at what point in the debilitation of the human brain does it finally depart? Speculative discussion pertaining to these matters is presented in Box 2-3.

Pleasure, Pain, and the Emotions. The emotions of pleasure and pain appear to be such deep integral parts of the human experience that it is difficult to believe that all that is happening within the brain is the firing of a few specific neurons. Yet, it can be demonstrated that in some sense this is indeed the case.

In experiments performed in 1939

[7]We use Penfield's terminology, even though it is now considered obsolete.

BOX 2-3 The Origins and Machinery of Consciousness

When and how did consciousness evolve, and where does it reside in the human brain? Three books addressing these questions all suggest that consciousness is a recent biological invention, closely linked to linguistic competence.

Julian Jaynes [Jaynes 77] offers the strange and somewhat unbelievable thesis that consciousness was first "invented" in Mesopotamia around 1300 B.C. He associates consciousness with the ability to think, plan, desire, hope, and deceive, and asserts that these attributes were lacking in early man and lower animals who were only capable of a stimulus-response pattern of behavior. He believes that the brain was originally organized into two functional components, an executive part called a "god" and a follower part called a "man," neither of which were conscious in the sense given above. Jaynes's main argument in support of his theory is that early man ascribed his actions to the inner voice of the god telling him what to do (e.g., Odysseus in the *Iliad*). Consciousness, according to Jaynes, was invented by man coming to the explicit realization that it is he, and not the gods, who directs his actions. With a different view of his thought processes, man's behavior itself changed from reflexive to introspective awareness.

Curtis Smith [Smith 85] argues that biological mechanisms created a linguistic capability before human language was invented, and that both language and consciousness are related evolutionary consequences of purely neurological developments. These critical biological inventions, specifically the development of a mental capacity for manipulation of information in the form of a general symbolic code, were required to integrate information from different sensory modalities* each describing the perceived world in a different "language." The evolutionary changes supposedly occurred with the emergence of Cro-Magnon man as a replacement for the prelinguistic preconscious Neanderthal man on the order of 50,000 to 100,000 years ago. (Neanderthal man, the first representative of our species, appeared approximately 150,000 years ago, but non-ape hominoids who made tools and used fire had already existed for more than 2½ million years.) Language allowed Cro-Magnon man to rise above the limitations of sensory experience, enabling him to possess an internal conscious world with the capacity to dream, imagine, remember,† and create.

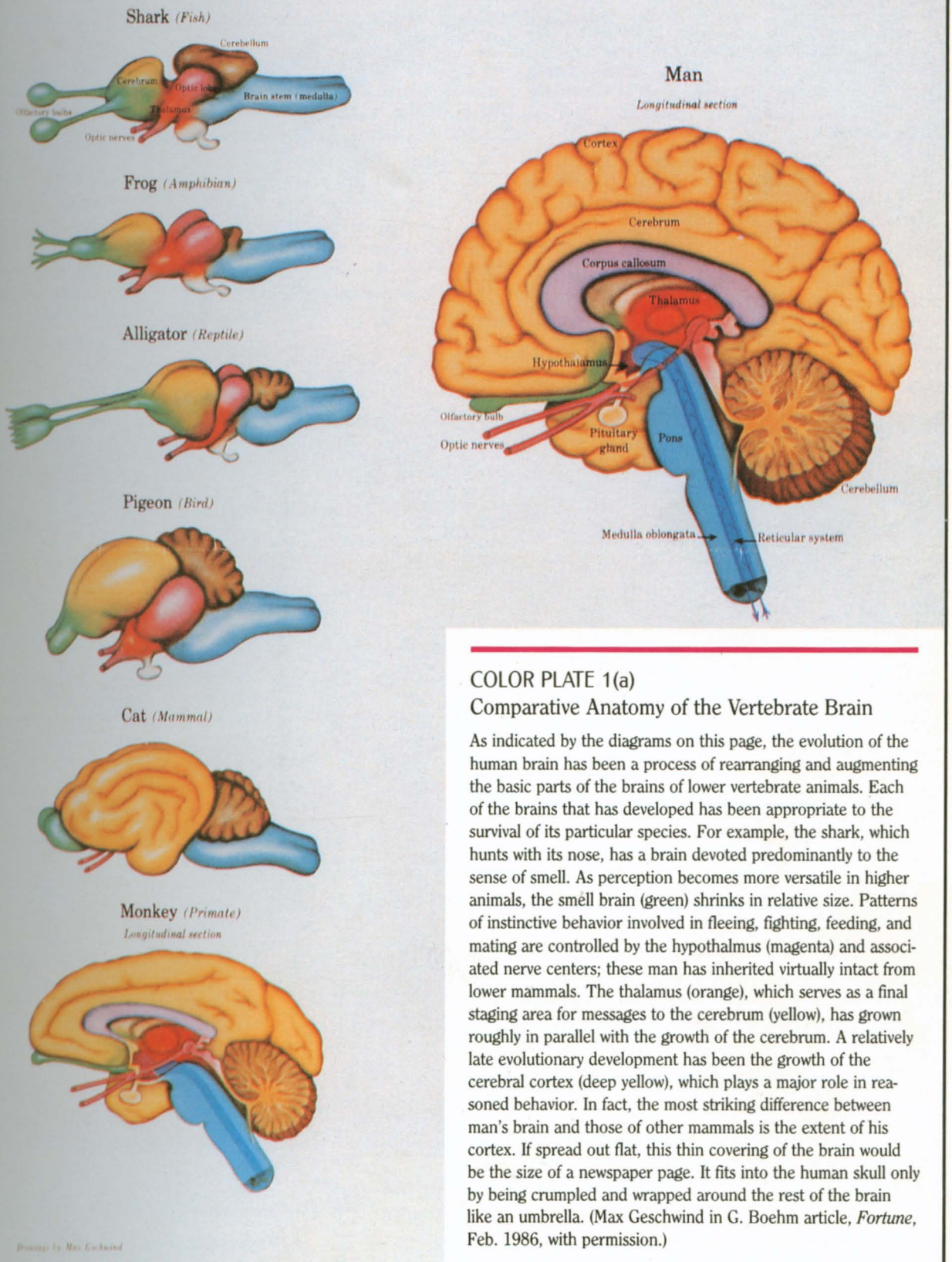
Michael Gazzaniga [Gazzaniga 85], a key scientist in the split-brain experiments described in Box 1-3, offers a unique and extremely provocative theory of consciousness. Like C.G. Smith, he believes that consciousness is only possible in man, and only developed after the evolution of both language and reasoning ability. However, he asserts that the brain is composed of multiple independent nonverbal modules and a single verbal module. The verbal module, which is the seat of consciousness, "observes" and attempts to explain the actions of the other modules:

> It has been commonplace to think that our conscious cognitive self is organized and exists in such a way that our language system is always in complete touch with all our thoughts. It knows where in our brains to find all information we have stored there, and it assists in all computations or problem-solving activities we engage in. Indeed, the strong subjective sense we all possess of ourselves is that we are a single, unified, conscious agent controlling life's events with a singular integrated purpose. . . . And it is not true. . . . There are a vast number of relatively independent systems in the brain that compute data from the outside world. These independent systems can deliver the results of these computations to the conscious verbal system, or they can express their reactions by actually controlling the body and affecting real behaviors.

Thus, according to Gazzaniga, conscious beliefs are explanations (devised by the verbal module) of the behavior of the independent entities constituting the brain viewed as a social system.

*Such an integrative ability is completely lacking in lower animals.

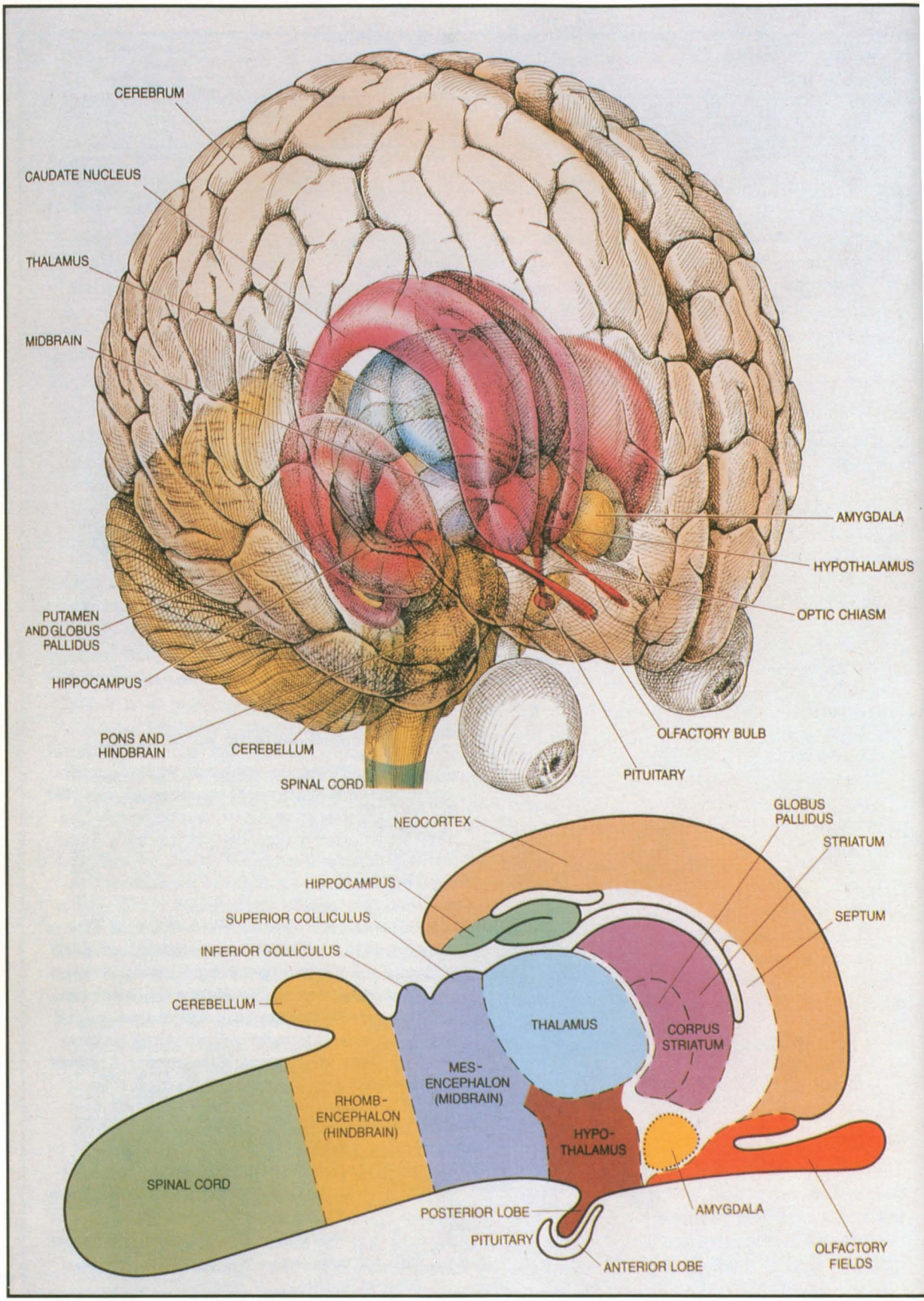
†Memory and consciousness are intimately related; in a sense, memory retrieval is consciousness. It follow that memory retrieval in lower animals that lack language must be a simpler and more sensory-oriented phenomena.



COLOR PLATE 1(a)

Comparative Anatomy of the Vertebrate Brain

As indicated by the diagrams on this page, the evolution of the human brain has been a process of rearranging and augmenting the basic parts of the brains of lower vertebrate animals. Each of the brains that has developed has been appropriate to the survival of its particular species. For example, the shark, which hunts with its nose, has a brain devoted predominantly to the sense of smell. As perception becomes more versatile in higher animals, the smell brain (green) shrinks in relative size. Patterns of instinctive behavior involved in fleeing, fighting, feeding, and mating are controlled by the hypothalmus (magenta) and associated nerve centers; these man has inherited virtually intact from lower mammals. The thalamus (orange), which serves as a final staging area for messages to the cerebrum (yellow), has grown roughly in parallel with the growth of the cerebrum. A relatively late evolutionary development has been the growth of the cerebral cortex (deep yellow), which plays a major role in reasoned behavior. In fact, the most striking difference between man's brain and those of other mammals is the extent of his cortex. If spread out flat, this thin covering of the brain would be the size of a newspaper page. It fits into the human skull only by being crumpled and wrapped around the rest of the brain like an umbrella. (Max Geschwind in G. Boehm article, *Fortune*, Feb. 1986, with permission.)


CEREBRUM
CAUDATE NUCLEUS
THALAMUS
MIDBRAIN
AMYGDALA
HYPOTHALAMUS
OPTIC CHIASM
PUTAMEN AND GLOBUS PALLIDUS
HIPPOCAMPUS
PONS AND HINDBRAIN
CEREBELLUM
SPINAL CORD
OLFACTORY BULB
PITUITARY
NEOCORTEX
GLOBUS PALLIDUS
STRIATUM
HIPPOCAMPUS
SUPERIOR COLLICULUS
SEPTUM
INFERIOR COLLICULUS
CEREBELLUM
THALAMUS
CORPUS STRIATUM
MES-ENCEPHALON (MIDBRAIN)
RHOMB-ENCEPHALON (HINDBRAIN)
HYPO-THALAMUS
SPINAL CORD
POSTERIOR LOBE
AMYGDALA
PITUITARY
OLFACTORY FIELDS
ANTERIOR LOBE

COLOR PLATE 1(b)

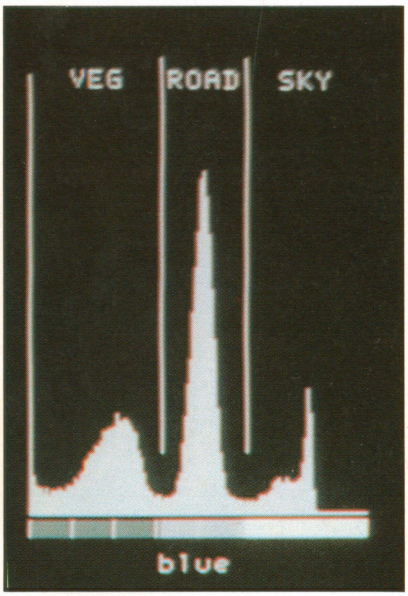
Comparative Anatomy of the Vertebrate Brain (*continued*)

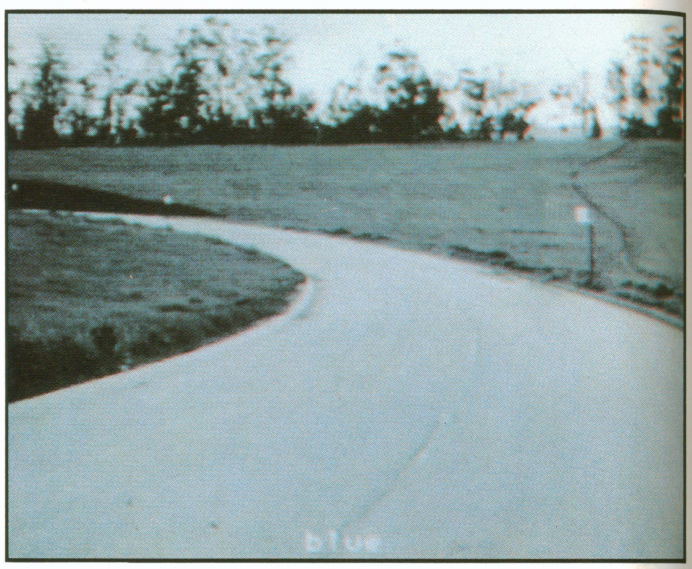
As indicated by the diagrams on this page, the evolution of the Brain and spinal chord of human beings and other mammals can be subdivided into smaller regions according to gross appearance, embyology, or cellular organization. At the top a human brain has been drawn so that its internal structures are visible through "transparent" outer layers of the cerebrum. At the bottom, a generalized mammalian brain is shown in a highly schematic view. Corresponding structures in the realistic and schematic models are the same color. The most general way of dividing the brain is into hindbrain, midbrain, and forebrain. The hindbrain includes the cerebellum. The midbrain includes the two elevations known as the inferior and superior colliculi. The forebrain is more complex. Its outer part is the cerebral hemisphere, the surface of which is the convoluted sheet of the cerebral cortex, which incorporates the hippocampus, the neocortex, and the olfactory fields. Within the hemisphere are the amygdala and corpus striatum, which includes the globus pallidus and striatum. The rest of the forebrain is the diencephalon: the upper two thirds comprises the thalamus (which has numerous subdivisions) and the lower third the hypothalamus (which connects to pituitary complex). (From W. Nanta and M. Feirtag, *Scientific American*, Sept. 1979, with permission.)

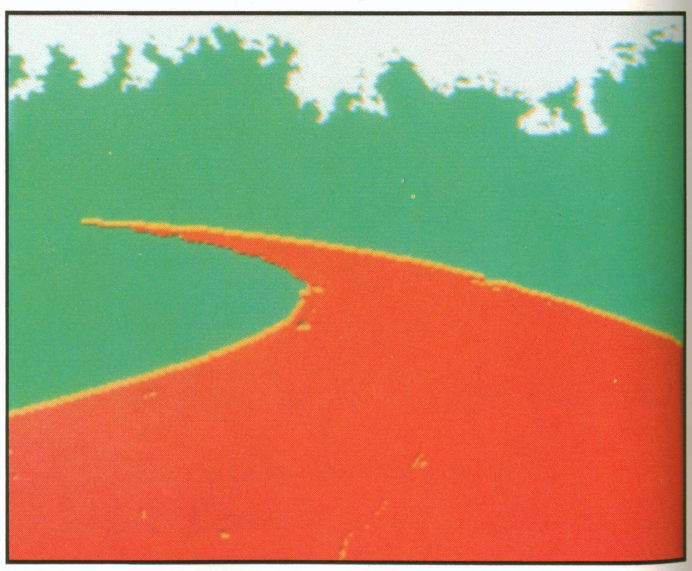
COLOR PLATE 2

A Synthetic Scene Generated using Fractal Textures.

(Fractal landscape rendering by R.F. Voss. From B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, San Francisco, © 1982 with permission.)







COLOR PLATE 3

Histogram Analysis for Automatic Threshold Setting.

Top right: Color image of a road scene.
Middle right: Blue component of color image.
Bottom right: Partitioned image, showing road, vegetation, and sky.
Above left: Histogram of blue component of color image, showing threshold settings.
(Photos courtesy of SRI International, Menlo Park, Calif.)

by Klüver and Bucy at the University of Chicago, it was discovered that when the region of the brain lying between the outer cortex and the center of the brain was damaged, monkeys behaved as if their emotional and motivational machinery was destroyed: they ate nuts and bolts as happily as raisins and randomly and inappropriately intermixed pleasure and fear responses to test situations.

More recent attempts to locate the emotional centers of the brain have narrowed the search to the hypothalamus, known (in addition to other functions) to control feeding, drinking, and sexual behavior. In 1953, James Olds discovered a region near the anterior hypothalamus of the rat that, when stimulated with an electrical current introduced through a "brain probe," provided such a high level of gratification that to get this reward rats would cross an electrified grid that previously had stopped rats starved for 24 hours from running for food.

While the positive response[8] to electrical stimulation of the hypothalamus has been demonstrated in rats, fish, birds, cats, dogs, monkeys, porpoises, and man, the interpretation of what is actually happening is not completely clear. In some cases it appears that the stimulation prevents termination or enhances the currently ongoing activity, rather than providing the subject with a pleasure reward. However, human subjects experiencing the positive effect, generally report that the stimulation caused reduction of anxiety or pain, or pleasurable feelings related to sex. One implication of these findings is that in spite of the complexity of human behavior, simple switches in the brain can turn on or off some of our strongest drives and motivating mechanisms.

Memory. Memory, nominally the ability to store and recall past events, is a critical component of human intelligence; after all, most of our reasoning deals with our previously stored knowledge of the world rather than exclusively with currently sensed data—defective memory is one of the most frequently observed symptoms of impaired brain function. What kinds of memory are there? How long can different kinds of things (a picture, a sound, a word, a story) be remembered? Does the human memory span exceed that of most other organisms? Is indeed memory simply a matter of storage and recall? Or is it a more complex function? What do memory defects tell us about the nature of normal human memory?

The first significant modern study of the psychology of memory was published by Hermann Ebbinghaus in 1885. He addressed such issues as the rate of forgetting (memory loss occurs quickly at first, then more slowly); "overmemorizing" and relearning ("each repetition engraved the material more and more deeply on the nervous system"); the amount of material that can be memorized (the learning time for n nonsense syllables is proportional to $n \log n$ for lists shorter than the immediate memory span); the effect of how the learning time is distributed (it is better to have several short learning sessions spaced out at intervals than to have one unbroken period of work), and a host of similar items.

Since memory was known to be strongly influenced by the meaning and novelty that the material has for the memorizer, the Ebbinghaus and most subse-

[8]Regions of negative response have also been found.

quent formal memory experiments attempted to achieve generality by employing nonsense syllables as data to be memorized. This approach masks the fact that, except in rare cases, the symbolic information memorized is an abstraction of the originally sensed data, rather than an exact copy. Thus, in normal situations memory is not simply a matter of storage and recall, but rather a complex process involving a considerable amount of cognitive processing.

The portions of the human brain thought to be involved with memory are the association areas of the frontal, parietal, occipital, and temporal lobes, and parts of the limbic system, especially the hippocampus. Little is known about the actual storage mechanisms and even less is known about the following ability which has no counterpart in computer memory systems: A person knows when something is stored in his memory, and when it is not. Thus, we will exert much effort to recall something that we "know we know," while we will make no effort to recall something that we know we do not know. For example, given the question, "What was Benjamin Franklin's telephone number?," we will not try to recall all of the telephone numbers that we know, but immediately conclude that no such number is stored in our memory.

Human memory is not a monolithic function—many different kinds of processes are involved and there are at least three[9] different types of memory: memory for sensed data, short-term memory, and long-term memory. The designation "short-term memory" is used to denote the ability to recall information presented a short time previously—short-term memory leaves no permanent imprint on the brain. One theory of short-term memory is based on the idea of "reverberation" of neuronal circuits in which an impulse travels through a closed circuit of neurons again and again. In this view, an incoming thought can be recalled while the reverberation continues. "Long-term memory," the indefinite retention of a memory trace, cannot be explained by reverberation. Rather, the concept of "facilitation" at synapses is used: when incoming information enters a neuronal circuit, the synapses in the circuit become "facilitated" for the passage of a similar signal later (triggered by some portion of the new signal which duplicates the original stimulus). Another theory suggests that long-term memory is related to protein synthesis by RNA: memory results from the production by RNA of specific proteins for each recorded event.

Each sensory modality (e.g., vision or speech) appears to incorporate a means of storing the complete incoming signal for on the order of 0.10 to 1.0 second. For example, we have all had the experience of not immediately understanding a spoken phrase, but by "replaying it" in our "mind's ear," we can recover the intended meaning. There are also visual "afterimages" which occur in a very short interval after the withdrawal of the stimulus, and are distinguished from other forms of visual memory in that these afterimages are not under voluntary control. We can inspect afterimages with our "mind's eye" and "see" things we did not observe

[9]There may be additional types of memory, e.g., Gazzaniga [Gazzaniga 85] describes evidence for the existence of memory mechanisms for storing procedural knowledge (such as motor skills) as distinct from mechanisms for storing declarative knowledge (facts or events).

when the visual stimulus was physically present.

In addition to very short-term sensory memory, there appears to be another form of short-term memory which lasts anywhere from 30 seconds to a few hours. Retaining a telephone number "in our heads" until we can complete the dialing —the number is typically forgotten almost immediately afterward—is an example of this type of memory.

Important information that is retained over long periods of time appears to be stored by a completely different mechanism from that used for the various types of short-term memory. But even here, more than one facility is involved. For example, there are memory disabilities in human patients that affect their ability to store and recall verbal material, while leaving intact their memory ability for nonverbal material.

Many other types of memory disorders are known that shed light on the multifaceted nature of human memory. For example, traumatic amnesia can be experienced by a person who has been knocked out by a blow on the head. In a confusional state lasting from days to weeks, the individual is unable to store new memories, and on recovery reports total amnesia for that period. Anterograde amnesia is the impaired ability to store memories of new experiences. (It is interesting to note that short term memory is typically intact among most amnesia sufferers. Some experimental psychologists believe that the primary factor in amnesia is the inability to transfer information from short-term to long-term storage.) Korsakoff's syndrome is a gross defect of short-term memory in which the sufferer may have access to memories of events occuring prior to the onset of the syndrome, but now immediately forgets each new experience; he lives only in the immediate present with no continuity between one experience and the next.

To summarize our main observation, except for very short-term sensory storage, the memory function is a complex activity that involves distinct modes of information partitioning, selection, and abstraction. It has all of the attributes of perception, and in fact, memory recall can be viewed as a form of internal perception. We do not generally retrieve a unique "token" in the exact form in which it was stored, but rather synthesize a "mental construct" (possibly from many different brain storage modalities) that is relevant to some purpose or ongoing process. The designation of perception, learning, and memory as distinct brain functions is a simplification which masks the true nature and interrelations of these activities.

Reasoning. Man has the ability to use current and past events to foresee possible futures, to plan and judge alternative courses of action, to deduce new facts from stored knowledge, and to reconstruct his environment from sensory data. Where and how does the human brain perform these functions which we ascribe to the general faculty called reasoning? It is in this particular matter that we least understand the machinery of the brain.

From a functional standpoint, we have already seen that reasoning is not a monolithic activity, but rather that there are at least two distinct paradigms the brain employs to solve the problems posed to it. The left hemisphere appears to be especially adept at solving problems

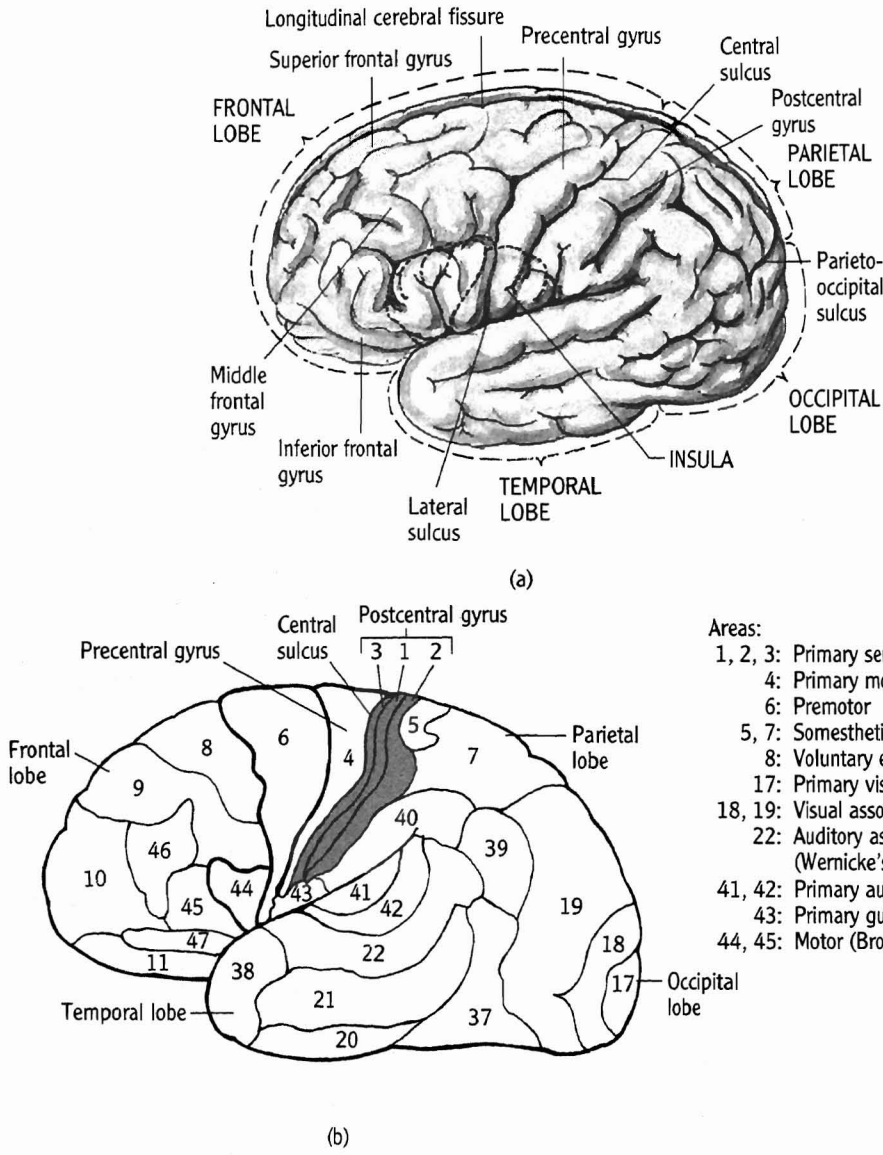


FIGURE 2-4 Functional Localization in the Human Cerebral Cortex.

(a) External anatomy of the cerebral hemisphere. Lobes, gyri, fissure, and sulci of the left cerebral hemisphere. The insula (indicated by broken lines) is hidden by the overlying lobes. (b) Functional areas of the cerebral cortex. The general area is shaded black. (From E. L. Weinreb. *Anatomy and Physiology*. Addison-Wesley, Reading, Mass., 1984, pp. 166,167, with permission.)

having a sequential/logical character, while the right hemisphere is superior in dealing with problems having a spatial/gestalt character. Other than the centers specialized to deal with data from specific motor and sensory systems (Fig. 2-4), no additional localization of the reasoning machinery is known. The computer analogy suggests that assemblies of neurons, individually capable of acting as both the logical switches and memory cells of a digital computer, indeed act as components in a distributed general-purpose computational engine executing relocatable algorithms recalled from memory as the need arises. There is no way at present to either verify or falsify this conjecture. In our current state of knowledge, we know as much (or as little) about reasoning in the brain as we do about the location and functioning of the human soul.

The Brain and the Computer. Before we move on to the computer portion of this chapter, let us examine where we have been and where we want to go. We studied the brain with the goal of learning more about intelligence. We discussed the physical structure of the brain, what the effects of damage are, and what we can introspect about human intelligence. Although these topics are of great intellectual interest, they do not provide the insights about intelligence that we originally hoped to attain.

We therefore turn to the computer with the expectation that, because we can analyze its structure and functioning in a way we cannot hope to do with the brain, we may be able to resolve some of our still-unanswered questions about intelligence.

THE COMPUTER

In its most basic sense, the computer is a machine that operates on information; it takes in information (or data) and transforms it in some specific way. As a physical device, the computer acts on physical quantities, and the assertion that it actually transforms information is an interpretation we impose on its behavior. Thus any physical system (the human brain, a dust cloud, a pocket calculator) is capable of being viewed as a computer.

To understand the behavior of a physical system viewed as a computer, and to determine what it is actually or ultimately capable of, a number of abstractions have been created that attempt to capture the essence of the concepts "computer" and "computation." It should be realized that the conclusions we draw by analyzing these abstractions—for example, conclusions about limits of performance—are valid assertions about the physical system only when viewed in the context of the abstraction; i.e., the limits are those of the abstraction. The most useful and powerful abstractions we have devised for formalizing the concepts of information, computer, and computation are based on the following two ideas:

1. The computer is an instruction follower.
2. The most complex set of instructions can be rewritten in a very simple language; i.e., a language which has an alphabet of only two letters (0,1) and a vocabulary of less than twenty distinct operations for altering strings of 1's and 0's.

The *Turing machine*, an abstraction based on these concepts, is described

later in this chapter. It will be seen that while the Turing machine does not lead to practical ideas about how to construct useful computers, it allows us to understand the limitations of all computer systems viewed as symbolic information processors (instruction followers that transform strings of symbols).

The Nature of Computer Programs and Algorithms

The *digital computer* (Appendix 2-2), the most widely used form of the computer, can be considered to be an instruction-following device, with the instructions presented in the form of a *program*. In most current computer systems, the hardware is controlled by a special internal program known as the *operating system*, which keeps track of how the computer resources are being used, and how the work is progressing. The user-provided programs to be processed are known as the *applications* programs. However, from the standpoint of the user, the separation between the hardware and the operating system is unimportant and often invisible; the combination forms the computer which "understands" instructions presented in one or more specialized languages.

Procedures must be described to an instruction follower in terms that are understandable to it. The instruction follower must be physically able to carry out the procedures, want to, or be made to, carry out the instructions in a practical amount of time, and be able to monitor progress and have a way of determining when the task has been completed. These requirements, assumed to be satisfied when we communicate with a person, must be explicitly met when communicating with a robot or computer.

Natural vs. Formal (Computer) Languages. Procedures are described to computers by means of *programming languages*, which have very precise rules of syntax and use. Such formal languages differ from *natural languages* such as English or French in the following ways:

- *Ambiguity*. A programming language is designed so as to avoid ambiguity; a single meaning can be found for each expression. On the other hand, natural language is often ambiguous: "I saw the orange truck."
- *Context dependency*. The meaning of a programming language expression is minimally dependent on its context; its meaning is almost always the same regardless of what its surrounding expressions are. In natural language, a sentence such as "I disapprove of your drinking" is changed in meaning when we add "so much milk."
- *Well-formedness*. In writing a programming language expression, one must follow the syntax rules exactly, otherwise the instructions will not be accepted by the system. In natural language, especially the spoken form, violations of syntax generally do not affect a person's comprehension of the expression.

Procedural vs. Nonprocedural Instructions. When presenting instructions to an instruction follower, we often specify both what we want done and specifically how we want the task carried out. This is known as providing *procedural* instructions. If, on the other hand, we indicate

what we want done, but do not specify "how," this is called a *nonprocedural* or declarative instruction. An example of a nonprocedural instruction is, "Buy a loaf of bread on your way home"; the desired end-effect is specified, but no specific instructions as to how to attain the goal are given. It is quite difficult to devise nonprocedural language systems, because the interpretation system within the computer must supply the 'how' by itself. The interpretive system must know about the "world" it is dealing with, and the effects of actions on this world. It must also understand the nature of the problem being solved. Using this knowledge, the system must devise a *plan*, a sequence of steps that must be performed to attain the goal. Because of these difficulties, only a few languages have been developed that have nonprocedural capabilities and these are generally limited to some specific domain.

Representation of Data in a Computer. A computer can be based on two distinct types of data representation: isomorphic or symbolic. In the "isomorphic" representation, data is modeled by a quantity which has a "natural" functional and possibly physical resemblance to the original data itself. For example, beads are used in the abacus to represent numbers, and the beads are physically moved to perform the computations. In the "symbolic" representation, the nature of the symbols used to represent the data is completely independent of the characteristics of the data being represented; the desired correspondence is established by a subsidiary set of rules. Thus, if we represent a number by its binary form, there is no natural relationship between the number and the form of its representation.

In current computer technology, the isomorphic representation is employed in the analog computer, which is fast and useful for dealing with certain physical problems, but has limited accuracy and flexibility. The symbolic representation is employed in the digital computer, which is extremely flexible and has unlimited numerical accuracy, but is comparatively slow and presents significant practical problems in accurately modeling many physical situations. For example, since the relationship between the physical situation and the computer representation is completely arbitrary, only those aspects of the physical situation that are both understood and can be described in a formal manner are capable of being modeled. Thus, a complete representation of an outdoor scene in a symbolic language would be an almost impossible task.

The Turing Machine

In order to prove formally what tasks can and cannot be performed by a computing device, Alan Turing, a British mathematician (1912–1954), postulated an abstraction, now called a "Turing machine," that is functionally equivalent to any computer. Turing's thesis was that any process that can be called an 'effective procedure' can be realized by his machine.

An effective procedure is a set of formal rules that tell a device from moment to moment precisely what operations to perform. (A computer program is an example of an effective procedure.) Turing's thesis cannot be established by

proof—it is actually a definition of the intuitive concept of a computable function, i.e., a function that can be evaluated by some finite algorithm. All attempts to define computability in some reasonable way have been shown to be equivalent to Turing computability.

In the Turing machine, the reduction of a process to elementary operations is carried to its limit. Even a simple operation such as addition is broken down into a chain of far simpler operations. This increases the number of steps in the computations carried out by the machine, but simplifies the logical structure for theoretical investigations.

As shown in Fig. 2-5, the Turing machine consists of a linear tape, assumed to be infinite in both directions, which is ruled into a sequence of boxes, or cells. The machine has a read/write head that can move from cell to cell of the tape, and can read or write symbols. At each moment of time, the machine is assumed to be in one of a finite number of internal "states" that are identified by the numbers 0, 1, 2, . . . The machine operation is controlled by a "state table" stored within the machine that specifies (1) the symbol to be "overprinted" at the current tape location (i.e., the old symbol is erased and a new symbol written), (2) the direction of head movement, and (3) the next state of the machine. The symbol printed, head movement direction, and the next state are determined by the current state of the machine, and the symbol that is on the cell of the tape currently being scanned.

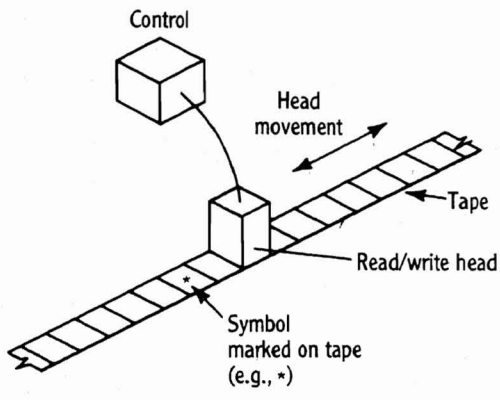


FIGURE 2-5 The Turing Machine.

The various operations that the machine can carry out are: the machine can halt; the previous symbol in a cell can be replaced by a new symbol; the read/write head can move one unit to the right or left; and the state number of the machine can be changed.

The state table 'instructions' are in the form of rows, each of which contains five elements: (1) old state, (2) symbol now being read, (3) symbol to be overprinted, (4) direction of head movement, and (5) new state. Thus, a state table entry [3,#,*,R,7] asserts, "If the old state is state 3, and the symbol being read is #, then the symbol * should be overprinted, the head should be moved one cell to the right, and the machine should go to the new state 7." Note that the first two symbols of a row cannot be the same as another row, since that would mean that there would be more than one operation specified for a given state and input symbol.

Turing showed that a state table could be prepared for each of the common operators such as addition, multiplication, and division, that more complex

operations could be composed from simple ones in a formal manner, and that his simple machine could carry out effective procedures equivalent to that possible on any computing device. Thus, anything proved about the ultimate capability of a Turing machine will hold for all computers. Box 2-4 shows a complete state table and how the corresponding Turing machine operates on a tape.

The Universal Turing Machine

It is possible to convert the entire state table of a Turing machine into a single number such that the original table can be recovered by decoding this number. A technique for performing this coding and decoding is shown in Box 2-5. We can thus say that the complete description of a Turing machine is given by its code number.

A "universal Turing machine" is a Turing machine that can take such a code number, decode it to obtain the state table of the original machine, and then execute that table. Thus, the universal Turing machine can simulate the operation of any Turing machine, given the code number of the machine.

LIMITATIONS ON THE COMPUTATIONAL ABILITY OF A LOGICAL DEVICE

A machine that operates on the basis of formal logical rules can be shown to have theoretical limits on its problem solving ability: there are certain well-posed problems for which no algorithm is possible, using the formal rules. In other words, we can prove that it is impossible for the machine to solve such problems!

The Gödel Incompleteness Theorem

At the beginning of this century, it was expected that mathematics would soon be mechanical in nature. Given a set of axioms and deduction rules, new mathematics would be produced by "blindly" applying the deduction rules to the ever-increasing set of mathematical truths. This mathematics would be consistent (no two statements produced would contradict each other), and it would be complete (every truth would be producible). Thus, one could eventually produce all true statements and never produce a falsehood.

This expectation was destroyed in 1931 by Kurt Gödel who showed that there are true statements in mathematics that a consistent formal system will not produce, i.e., that it is impossible to alter the foundations of mathematics to exclude unprovable propositions. Gödel showed how to produce a true statement, S, that could not be proved by a consistent system, F, using a set of axioms and a proof procedure. He did this by showing that if S could be proved, then a contradiction would arise. F is therefore "incomplete" since it does not produce all true statements. The approach is a formal treatment of the "liar's paradox":

Given the statement S: *This statement is a lie.* Then if S is true, S is false, while if S is false, then S is true. Gödel's approach used the form:

S: *This statement is not provable.* Then if S is provable, S is not true, and our formal system has produced a falsehood. If S is not provable, then we have a statement that is true, but not provable in the system, and the system is incomplete.

Gödel's approach to proving the

BOX 2-4 Programming a Turing Machine: The Parity Problem

Programming a Turing machine with "alphabet" [0,1] consists of preparing a control table that will cause it to operate on a binary input tape in a desired manner. Numbers on the tape can be represented as strings of "1" marks (the number 5 = "11111"), and if we have two numbers we can separate them by a cell that has a zero in it, e.g. 2,5 would be represented as ". . .001101111100. . .". Adding two numbers consists of preparing a control table that removes the zero between the strings; subtraction consists of shuttling back and forth between the two strings, stripping off 1's alternately for each until no 1's remain in the smaller string. Multiplication of a string m 1's long by one n 1's long consists of replacing each 1 of n by m 1's.

We will assume that the machine must shift left or right after performing an overprint and prior to entering its next state. The machine is always started at a specific position on the tape in control state 1. If the machine enters the zero state, it halts without performing any further operations.

The parity problem discussed below requires the machine to determine whether there is an even or odd number of 1's on a tape. Conceptually, a control table could be set up so as to toggle between the two states as the head moves along the tape. When a zero cell is encountered, the machine reports in some specified manner and halts. We will examine several forms of the problem.

Parity Problem (P1)

1. Input consists of string of consecutive 1's with start position at the rightmost 1 of string. The machine starts in state 1.
2. For odd parity (odd number of 1's in input string) we require the machine to stop at the leftmost 1 of the input string as shown in the example below:

```
input string  . . . 0011100 . . .
                        ↑
output string . . . 0011100 . . .
                      ↑
```

3. For even parity (even number of 1's in input string) we require the machine to stop under the second 0 to the left of the input string as in the example below:

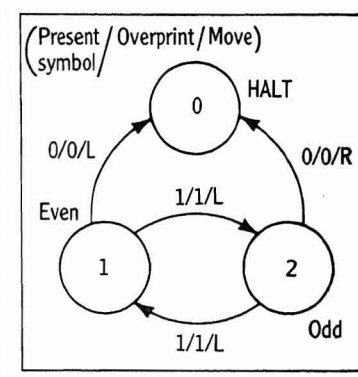
```
input string  . . . 001100 . . .
                      ↑
output string . . . 001100 . . .
                    ↑
```

A control table to solve the above parity problem is shown in Fig. 2-6. It is obvious that the P1 parity problem cannot be solved by a

FIGURE 2-6 Turing Machine for Solving the Parity Problem.

Control state	Present symbol	Overprint	Move	New state
1	0	0	0 (left)	0 (halt)
1	1	1	0 (left)	2
2	0	0	1 (right)	0 (halt)
2	1	1	0 (left)	1

(a)



(b)

BOX 2-4 (*continued*)

machine with a single control state. Thus, the above machine is optimal since it solves the parity problem with the minimum possible number of control states.

Now consider an apparently trivial variation of the parity problem in which we merely change the reporting requirements:

Parity Problem (P2)

1. For odd parity, print a 1 on the second square to the left of the input string and stop at this location:

```
input string     ...0011100...
                        ↑
output string    ...001011100...
                      ↑
```

2. For even parity, print 1's in the second and third squares to the left of the input string and stop at the second square:

```
input string     ...001100...
                      ↑
output string    ...001101100...
                    ↑
```

Try to find a machine that will solve the P2 problem. It should be easy to find a six-state machine; if you are very clever and are willing to spend a lot of time, you may even find a four-state machine [Rado 62].

Nobody has yet found a three-state solution to P2, nor do we know if one exists. It might seem feasible to resolve the issue of the existence of a three-state machine by writing a conventional computer program to exhaustively try out all possibilities. We note that there are more than 16 million three-state machines, and for each such machine we would have to determine if it will report parity correctly, and halt, for every possible input string. As discussed earlier in this chapter, and in Box 2-6, even deciding whether a machine will halt, given an all-zero input tape, is a generally undecidable problem. Thus, while intuitive or heuristic search techniques could conceivably produce a three-state solution to the P2 problem, failure to find such a solution does not imply that one does not exist, nor do we have a formal method, at present, to resolve this issue.

general existence of unsolvable problems has subsequently been used to show that specific problems are unsolvable. For example, Hilbert's tenth problem, one of the famous problems of mathematics, has been shown to be unsolvable. (This problem is to find a general algorithm that could determine in a finite number of steps whether or not a given Diophantine equation has an integer solution.) This has been shown to be unsolvable by using a proof that involves Gödel numbering of a statement related to Diophantine equations, and demonstrating the Gödel contradiction.

Unsolvability by Machine

The Gödel concepts carry over into machine unsolvability, since once we have the idea of a single number representing an entire machine (see "The Turing Machine" above), we can prove theorems about unsolvability. For example, Box 2-6 presents an informal proof for the *halting problem*: there cannot be a machine X that when given the state table of an arbitrary machine Y and its starting tape, is able to tell whether machine Y will ever stop. Box 2-7 discusses the *busy beaver problem*, which demonstrates noncomputability. Other examples of unsolvable problems for a Turing machine (thus any computer) are:

- *Machine equivalence.* It is impossible to have a machine that, given the state tables of any two Turing machines, S and T, always can tell whether S is equivalent to T.

BOX 2-5 Gödel Coding: Coding a State Table into a Single Number

There are many ways of coding a sequence of numbers into a single number so that the number can be uniquely decoded back into the original sequence. The one used by the mathematician Kurt Gödel in his original work on undecidability is based on prime factors. If we have a sequence of non-zero integers S = a, b, c, . . . n_k, we can form the product, $N=(2^a)(3^b)(5^c) \ldots (p_k^{nk})$, where 2,3,5, . . . p_k are all prime numbers, and p_k is the k^{th} prime number. N is then the code number of the original sequence, and since N was formed from the product of primes raised to a power, we can uniquely determine the power of each prime and hence can recover the original sequence of numbers.

Given a control table of a Turing machine, we first convert all the entries to numbers and eliminate any zeros in the table:

- Right, left, and halt are denoted by some numbers, say 1,2,3, respectively
- Any symbols to be printed are represented by a number
- If any state is labeled as 0, add 1 to all states

We now have a set of five numbers in each row of the control table, and we can concatenate rows to form a long sequence of numbers. The sequence can be converted into a single number using the prime number encoding approach described above. We can then talk about the Turing machine N, meaning the control table of the machine coded into the number N using Gödel coding.* As an example of this, consider the control table used in Box 2-4. We first eliminate the 0 values by adding one to all numbers. We then obtain a Gödel number for each row:

$10000 \rightarrow 21111 \rightarrow 2^2\times3\times5\times7\times9 = A$
$11102 \rightarrow 22213 \rightarrow 2^2\times3^2\times5^2\times7\times9^3 = B$
$20010 \rightarrow 31121 \rightarrow 2^3\times3\times5\times7^2\times9 = C$
$21101 \rightarrow 32212 \rightarrow 2^3\times3^2\times5^2\times7\times9^2 = D$

If we call the row Gödel numbers A, B, C, and D, then we can code the entire table into the number $N=2^A\times3^B\times5^C\times7^D$. Since N encodes the original control table, we can then use the designation "machine N."

*Note that this coding approach is conceptual, rather than practical, since the product of the primes raised to a power is an impractically large number.

- *Symbol prediction.* It is impossible to have a machine that can determine whether an arbitrary machine A will ever write the symbol S when started on tape B.

Implications of Gödel's Theorem

Gödel's theorem, showing that in any formal system there are true statements that are unprovable in the system, has had a profound effect on the philosophy of mind. Some see the theorem as indicating a basic limitation on both human and machine intelligence, while others see the human as somehow escaping the Gödelian limitation. There is also the view that Gödel's theorem has little relevance to the issue of achieving intelligent behavior. The arguments are as follows:

Man and machine limitation. Both people and machines consist of "hardware" that operates according to strict mechanical laws. In the case of computers, the electronic mechanism constitutes the formal system, while for the human, the formal system is the neural structure. Therefore, there will be truths unknowable by both man and machine.

Only machine limitation. People are not machines—they exceed strictly mechanical limits by their ability to introspect and to interpret experience. The powers of mind exceed those of a logical inference machine.

Neither man nor machine. The importance of proof techniques in consistent systems has been overrated. Most of our knowledge about the world comes from inductive methods that operate in inconsistent systems. Gödel's incompleteness theorem simply places a limit on one mode of obtaining new knowledge.

Computational Complexity—the Existence of Solvable but Intrinsically Difficult Problems

We have already observed that there are some well-posed problems in mathematics and logic for which no algorithm can ever be written (e.g., the halting problem)—thus there exists a set of theoretically unsolvable problems. However, even for problems that have solutions, there is a subclass of intrinsically difficult problems —problems for which there cannot exist an efficient algorithm. Intrinsically difficult problems are characterized by the fact that their solution time grows (at least) exponentially with some parameter indicative of problem size (e.g., the number of Turing machine control states in the case of the busy beaver problem discussed in Box 2-7). Such intractable problems often arise from the need to exhaustively search a solution space which grows exponentially with problem size; many optimization problems for which no solution space gradient exists (and can only be solved by the equivalent of a "backtrack" search algorithm) have this characteristic. Thus in addition to theoretically unsolvable problems, we also have a class of computationally unsolvable problems.

Between those problems for which we have efficient (polynomial time[10]) solutions, and those problems known to be intractable, there exists a large class of problems with the following interesting set of characteristics:

(a) There is no currently known polynomial time sequential algorithm for any of these problems; we suspect that they are all intractable, but cannot prove it.

(b) They are all equivalent to the satisfiability problem (Does a given Boolean or logical expression have an assignment of its variables that makes it "true?" See Chapter 4). If a polynomial time algorithm could be found for any one of these problems, then they could all be solved with polynomial time algorithms.

(c) While the size of their solution space grows exponentially, the number of operations needed to find a solution to any of these problems is polynomial if we choose all the correct alternatives. Thus, with enough computers running in parallel, each checking a different alternative at each decision point, we can achieve polynomial time solutions with an exponentially large amount of hard-

[10]The number of computational steps needed to assure a solution is expressible as a polynomial in one or more of the main variables in which the problem is posed.

BOX 2-6 The Halting Problem

The following dialogue is an informal proof of the impossibility of having a machine that can tell in general whether another machine will ever halt.

John: I've written a program called TESTER that tells when another program has an endless loop in it.

Mary: How does it work?

John: I have a way of uniquely assigning a number that represents an entire program. For example, if you give me a program, I first compute the number of the program. Then you give me the number that you would input to that program. Suppose I am given a program whose number I find to be 397 and you want to know whether it will halt if you feed in the number 14 to it. I feed these two numbers into TESTER, and if program 397 would halt given an input 14, then TESTER will output a 1, otherwise it will output a 0. TESTER has the form (see Fig. 2-7a):

```
TESTER(N,D) :
If Program N would halt on input D,
RETURN 1;
else return 0
```

Mary: Is it O.K. for me to write the following program? (see Fig. 2-7b):

```
        TRYOUT(X) :
Label L: If TESTER(X,X) = 1
        then go to Label L;
        else RETURN X
```

John: TRYOUT seems to be O.K. It says that if TESTER using the program numbered X and input data X causes TESTER to output a 1, then TRYOUT will loop endlessly to L, otherwise TRYOUT returns X.

Mary: Does TRYOUT have a number?

John: Of course, every program has a number. Let's see, it comes out to 4,396. So TRYOUT(4,396) would say that if TESTER(4,396, 4,396) = 1 then go to L; else RETURN (4,396).

Mary: I think that something is wrong. If the output of TESTER is 1, then the program being tested is a program that would halt.

John: Right.

ware. Further, for all of these problems, if we could somehow guess the correct answer, we could check the validity of the answer in polynomial time.

This class of problems, called the *NP*-complete class, includes such well-known problems as the "traveling salesman problem" (find the shortest closed route over a given set of roads that passes exactly once through each of a given set of cities) and the "Steiner minimal tree problem" (design the shortest network of roads that connects a given set of cities).

The existence of intrinsically difficult problems indicates the need to employ representations and algorithms that can find approximate solutions, i.e., representations that embody the concept of distance to a solution. We note that some of our most powerful "exact" techniques (such as the logical formalism, Chapter 4) do not have a natural way of representing solution space distance.

An important question left unanswered in this subsection is whether the

BOX 2-6 (*continued*)

Mary: But if TESTER outputs a 1 when using my program TRYOUT having a number 4,396, that means that TESTER thinks that my program should halt.

John: Right.

Mary: But look at my program. With a 1 output from TESTER, my program loops! What's more, if TESTER outputs a 0, that means my program doesn't halt, but if you look at my program you see that with a 0 output from TESTER my program halts (see Fig. 2-7c).

John: That seems a lot like the paradox, "This statement is a lie." If the statement is true, then it's false, and if it's false, it's true.

Mary: That's right, this proof of the halting problem uses that general approach.

A rigorous treatment of the above Turing machine proof can be found in Minsky [Minsky 67].

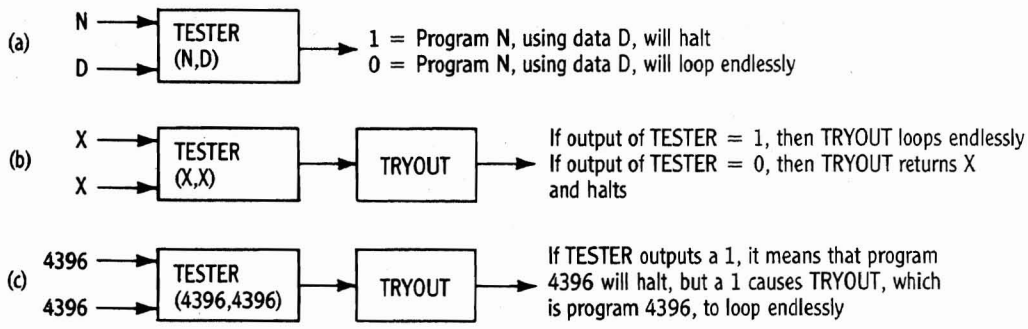


FIGURE 2-7 Programs for Illustrating the Unsolvability of the Halting Problem.

(a) Program TESTER. (b) Program TRYOUT. (c) The paradox.

classification of a problem as tractable or intractable[11] is a function of the representation employed. This question can be answered for the case of Turing machine equivalent computers—any reasonable problem encoding does not alter tractability—but what happens if we employ an analog device that is not equivalent to a Turing machine (see Box 2-8)? While we do not yet know the answer, it is interesting to observe that no analog solutions have been found for intractable problems.

[11]Whether the problem has a polynomial time solution or not.

LIMITATIONS ON THE COMPUTATIONAL ABILITY OF A PHYSICAL DEVICE

From the day of birth, and probably before, but certainly every day afterward, upward of 1000 neurons die in the human brain and are not replaced. How can the

BOX 2-7 Nonsolvability, Noncomputability, and the Busy Beaver Problem

It has been known for some time that unsolvable problems exist within specific mathematical systems. For example, it can be shown to be impossible to trisect an arbitrary angle using only a straightedge and compass. There are also undecidable questions: for example, Lobachevsky proved that the parallel postulate in geometry is independent of Euclid's axioms, and thus, neither it, nor its negation, can be proved within a Euclidean system. While the above specific examples are easily dealt with by extending the axiom systems in which these problems are embedded,* there are also problems that are absolutely unsolvable in the sense that there is no finite algorithm for dealing with them. Such problems, first introduced by Church, Gödel, and Turing, are called "recursively unsolvable."

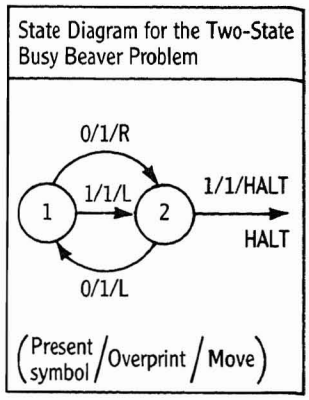
There is a close relationship between the incompleteness theorems of Gödel and the noncomputability results of Turing. Both rely on a form of Gödel coding to make self-referring statements in a modified version of the "liar's paradox"; the proofs are then established by contradiction. While mathematically sound, these methods do not provide an intuitive explanation as to why, for example, there should be well-defined numerical values that cannot be computed, or the relation-

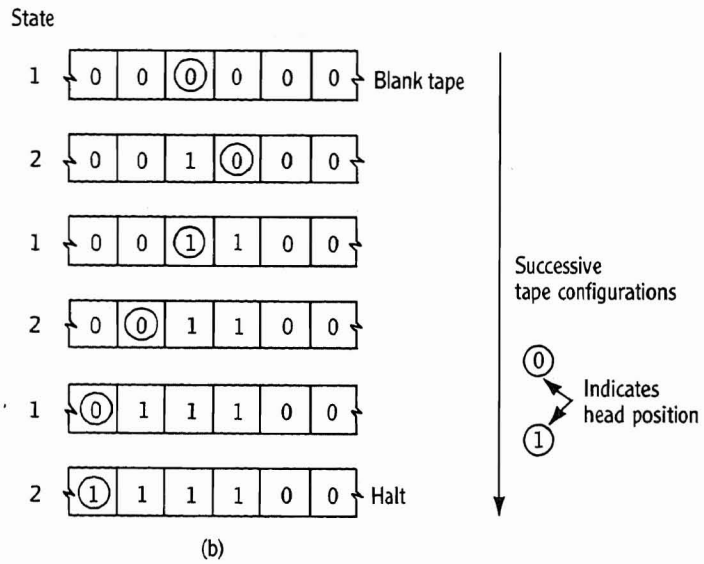
*Though, as Gödel showed, such extensions introduce new undecidable problems

State Table for the Two-State Busy Beaver Problem

Control state	Present symbol	Overprint	Move	New state
1	0	1	R	2
1	1	1	L	2
2	0	1	L	1
2	1	1	Halt	—

(a)





(b)

FIGURE 2-8
Turing Machine Solution for the Two-State Busy Beaver Problem.

(a) State table and state diagram. (b) Starting with a blank tape, the machine writes four 1's and then halts. (After A. K. Dewdney. *Scientific American* 251:19-23, 1984.)

BOX 2-7 *(continued)*

ship between noncomputability and intrinsically difficult problems. The busy beaver problem discussed below will clarify some of these issues.

Consider the problem of determining the maximum number of 1's that can be written on an initially blank tape by a Turing machine (Box 2-4) having an n-state control table. We will call this the busy beaver-n problem, and will let $b(n)$ designate a solution machine and $g(n)$ the corresponding number of 1's. We will only consider as valid solutions machines that halt after writing their tape. Figure 2-8 shows a two-state busy beaver machine that begins with a blank tape, writes four 1's, and then stops.

It can be shown that there are $m(n) = [4(n+1)]^{2n}$ machines having n states, and since we must examine each machine to determine $b(n)$ and $g(n)$, then the busy beaver problem is intrinsically hard since $m(n)$ grows exponentially with n.

For example, we have:

$$m(1) = 64$$
$$m(2) = 21(10^3)$$
$$m(3) = 17(10^6)$$
$$m(4) = 26(10^9)$$
$$m(5) = 63(10^{12})$$

Thus, $m(n)$ is of astronomical size, even for low values of n. The number of 1's, $g(n)$, ultimately grows at a much faster rate—in fact, it can be proved that for any computable function $f(n)$ there is a value of n beyond which the value of g exceeds that of f. Since g grows faster than every computable function, $g(n)$ cannot be computed; i.e., a finite algorithm cannot be formulated that will produce correct values for $g(n)$.

For small values of n we can explicitly evaluate g as shown below:

$g(1) = 1$ $\quad$ $g(5) < 17$
$g(2) = 4$ $\quad$ $g(6) < 36$
$g(3) = 6$ $\quad$ $g(7) < 23{,}000$
$g(4) < 14$ $\quad$ $g(8) < 10^{42}$

Intuitively, it appears that we can ascribe noncomputability (at least in the above case) to the inability of finite algorithms, based on primitive arithmetic operations, to express all possible functions, especially those with a sufficiently fast rate of growth. However, we cannot ignore the fact that the busy beaver problem includes the halting problem (i.e., we must examine both the number of 1's produced by every potential solution machine and also assure ourselves that it stops), and the halting problem again implies the presence of the Gödelian paradox [Jones 74].

brain continue to function under such conditions, since the loss of even a single component in a modern digital computer will typically render it inoperative? Even more to the point, some biological mechanisms appear to be deliberately designed to take advantage of failure and error in their physical components—one such example is the paradigm for evolution employed by DNA (mutation and natural selection). It is believed that DNA actually adjusts its error rate to produce a percentage of mutations appropriate for current environmental conditions. Under less favorable conditions, a higher rate of mutation has an improved survival value for the species.

Reliable Computation with Unreliable Components

Below a certain level of complexity, things tend to break down—to become more random in their organization. This observation is important enough to have been elevated to a basic law of physics (the second law of thermodynamics). However, very complex systems can be organized so that in spite of the breakdown of their individual components, they continue to function; most living organisms have this characteristic.

In 1952, John von Neumann showed that if the neurons of the brain could be considered to behave as logical switches,

BOX 2-8 Avoiding the Apparent Bounds on Computational Complexity

The Turing machine is more than an abstraction of the digital computer, it is actually a formalization of the sequential logical paradigm—in a sense, it can be taken as an abstraction of the conscious mode of human thinking. Thus it comes as a surprise that complexity bounds, derived for sequential algorithmic computation, can be violated by using a different underlying representation.

For example, consider the problem of sorting a set of numbers. In order to put the numbers into sequential order, the basic operation to be performed is that of comparison, and it has been proved that at least "on the order of" $n\log n$ comparisons are required to sort n numbers; i.e., that computation time must grow faster than a linear function of n. Now consider performing the same sorting task on the "spaghetti computer" [Dewdney 84]. We first cut n pieces of uncooked spaghetti so that each piece has a length proportional to one of the numbers to be sorted; this requires a time proportional to n. Next, loosely holding all the cut pieces of spaghetti in one hand, bring the bundle sharply down on a flat horizontal surface, thus aligning the ends of all the pieces of spaghetti—a single operation. Finally, obtain the desired sorted sequence by first removing the tallest (most protruding) piece; then the tallest of the remaining pieces, and so on until the bundle is exhausted. As each piece is removed from the bundle, it is measured and the resulting number is recorded; this set of operations is linear in n. Thus, the entire sorting operation done on the spaghetti computer requires a series of linear time operations and can be accomplished in linear time—violating the $n\log n$ computational bound on sorting derived for sequential machines.

Two additional examples of how computation in an appropriately chosen analog (isomorphic) representation can violate a bound on sequential computation are:

(a) The "convex hull" of n points is the smallest convex region containing all n points. The convex hull is a polygon, each of whose vertices corresponds to one of the extreme points of the set of points. While there is a $n\log n$ bound on finding a planar convex hull, this sequential machine bound can be violated by using the "rubber band computer" (Fig. 2-9). When stretched to fit over all the pins and then released, the rubber band will form the convex hull of the points.

(b) The problem of finding the shortest path joining two selected vertices of a graph has a sequential machine complexity of order n^2. We can violate the

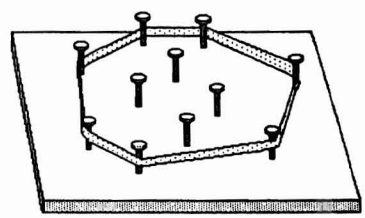


FIGURE 2-9
The Rubber Band Computer.

This computer determines the convex hull of a planar set of points. (After A. K. Dewdney. *Scientific American* 250:19-26, 1984.)

as in a computer, then an arbitrary degree of failure tolerant operation could be achieved at a cost of massive redundancy or repetition; i.e., by employing switches/neurons wired in parallel and performing the same function (see Box 2-9). In 1948, Claude Shannon proposed a more sophisticated scheme for using redundancy in the context of achieving reliable transmission or storage of information. He showed that rather than just repeating the message many times, it was more efficient to encode the message so that each valid message had no close "neighbors" in "message space." Thus, if a message was slightly altered by noise or transmission

BOX 2-8 (*continued*)

n^2 bound on finding a shortest path in a graph using the "string computer" (Fig. 2-10). Each vertex is represented by a ring or knot, and if two vertices are joined by an edge, the corresponding rings are connected by a piece of string cut to the correct length and tied to the rings. To find the shortest path between vertices *a* and *b*, pick up the network by the rings *a* and *b* and pull the network taut. The shortest path is the sequence of taut strings. (As an interesting aside, if we pull hard enough to break the strings, the last set of strings that retains the connectivity between rings *a* and *b* is usually the longest path between these rings. It can also be shown that the longest path between any two vertexes can be found by first picking up the tree by any ring, and then holding the tree by the lowest dangling ring; the longest path runs from the ring being held to the one that now hangs lowest [Dewdney 85].)

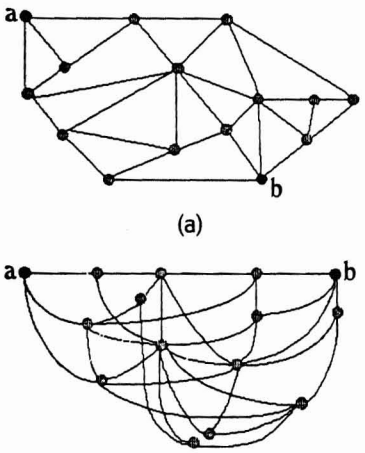


FIGURE 2-10
The String Computer.

This computer finds the shortest path between two specified vertices in a graph. (a) String analog of a given network. We are required to find the shortest path between the two darkened vertices. (b) The solution path found by grasping the selected vertices, and pulling in opposite directions.

While no examples of analog computation are known to provide a complete effective solution to intrinsically difficult problems, the "soap film computer" (Fig. 2-11) can find individual potentially optimal solutions to the *NP*-complete Steiner minimal tree problem in linear time. The Steiner-tree problem asks that *n* points in the plane be connected by a graph of minimum overall length. One is allowed to take as vertices of the graph not only the original *n* points, but additional ones as well. The soap film computer consists of two sheets of rigid plastic with pins between the sheets to represent the points to be spanned. When this device is dipped into a soap solution, the soap film connects the *n* pins in a Steiner-tree network.

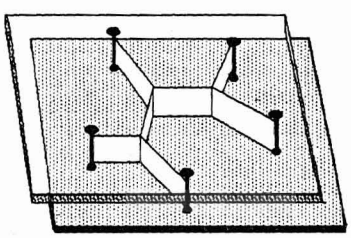


FIGURE 2-11
The Soap Film Computer.

This computer finds the solution for the shortest path connecting a planar set of points (the minimum Steiner-tree problem). (After R. Courant and H. Robbins. *What is Mathematics?* Oxford University Press, London, 1941.)

error, the resulting message would likely correspond to a point in message space that was near the original message but which itself did not correspond to another valid message. The original message could then be recovered from a received erroneous message by finding the nearest valid message in message space. Figure 2-13 shows the "space" of all conceivable messages, and the legal messages are indicated as distinguished points in that space. A message containing an error will not coincide with any of the distinguished points, but if it lies within the shaded sphere surrounding a legal message, then it is assigned to that legal message.

BOX 2-9 The Use of Redundancy to Achieve Fault Tolerant Computing

We are so used to the idea that the transcription of symbolic information and the operations performed in mathematics and logic can and must be error free, that it is easy to lose sight of the fact that perfection is almost never present in the physical artifacts that man constructs.* Yet, machines capable of formal symbolic computation must be perfect in the way they represent and transform information. As we will see in our discussion of logical reasoning in Chapter 4, such perfection is the essence of a strategy for dealing with complexity. The machines we build must employ some other strategy to attain perfection with imperfect components.

It is possible to obtain a reliable computing system using components that are subject to failure by using *redundancy*, i.e., more components than are necessary to accomplish the task. In addition to redundancy, it is also necessary to employ a connection or control scheme that takes into account the nature of the computation and the failure modes of the components. Redundancy can be utilized at various levels in the design hierarchy: at the level of the single component, at the subsystem level, and at the level of completely functioning systems. Two of the many different approaches to fault-tolerant computing are described below.

*Haugeland [Haugeland 85] in a relevant discussion contrasts the fate of Rembrandt's paintings, which are slowly deteriorating, with Shakespeare's sonnets which, as symbolic constructs, can be preserved exactly the way the author wrote them.

Component Replication

Suppose we have a simple component, designated as ▶|, that permits electrical current to flow in one direction, but not in the other. Thus a circuit, A—▶|—B would permit current to flow from A to B but not from B to A. The ▶| component can fail "open" and not permit any current flow, or fail "shorted" and allow the current to flow in either direction. We want to design a circuit that operates properly despite these types of failure. Note that placing two or more ▶| in parallel will not solve the problem, because a short in any ▶| will cause the circuit to fail. Instead, we must use the "series-parallel" circuit shown in Fig. 2-12, which can operate properly even though a short has occurred in a single ▶| in all of the N parallel paths. It can also operate properly even if (N−1) of the paths contain open ▶| elements.

In the 1950s, the mathematician John von Neumann showed how reliable organisms could be synthesized from unreliable components. Since in a complicated network the probability of errors in the basic processors could make the response of the final output unreliable, he sought some control mechanism to prevent the accumulation of these errors. The approach that he developed, similar in concept to that shown in Figure 2-12, uses N redundant paths for all operations. Thus, each input line is replicated N times and fed into N identical computing elements; this replication continues throughout the entire system. At the output of the system we have N output lines, and the final result will be accepted if a certain percentage of the output lines agree. For example, in a biological system, the N parallel outputs could be N distinct muscle strands comprising a single muscle. The muscle will flex if a certain percentage of strands agree.

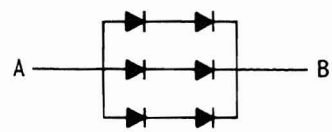


FIGURE 2-12
Redundancy Achieved by Component Replication.

Duplicative redundancy is innately inefficient. For example, suppose we have unreliable computing elements with failure rates of one failure every 200 operations. Using the von Neumann approach, a computing machine with 2500 such elements being actuated every five microseconds would require replication by a factor of 20,000 to obtain eight hours of error-free operation!

Cooperating Redundant Systems

If a set of processors has sufficient freedom to communicate, then we can develop a reliable system whose

BOX 2-9 (*continued*)

operation corresponds to a group of people working jointly on a problem. Certain controls must be incorporated into the system so that there is an effective way of partitioning the work, and so that a deviant processor does not write into the memory of another processor, does not tie up the communications channels, and does not seize the output mechanism. Processors can report to one another concerning their opinion on the "health" of any of the processors, and processors can ignore and redistribute the work of a processor that a consensus of the processors believes is unreliable. The type of "software implemented" fault tolerance has been used as the basis for computer systems that are required to have high degrees of reliability.

An example of message error detection and correction is presented in Fig. 2-14. Figure 2-14(a) uses a Venn diagram to show how three parity bits can provide single error detection and correction of a four-bit message. Figure 2-14(b) shows a code based on this concept.

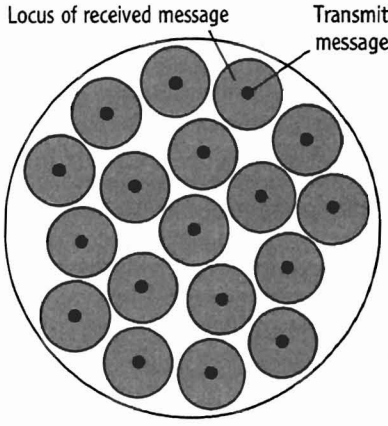


FIGURE 2-13
Message-Space Representation of a Fault-Tolerant Coding Technique.

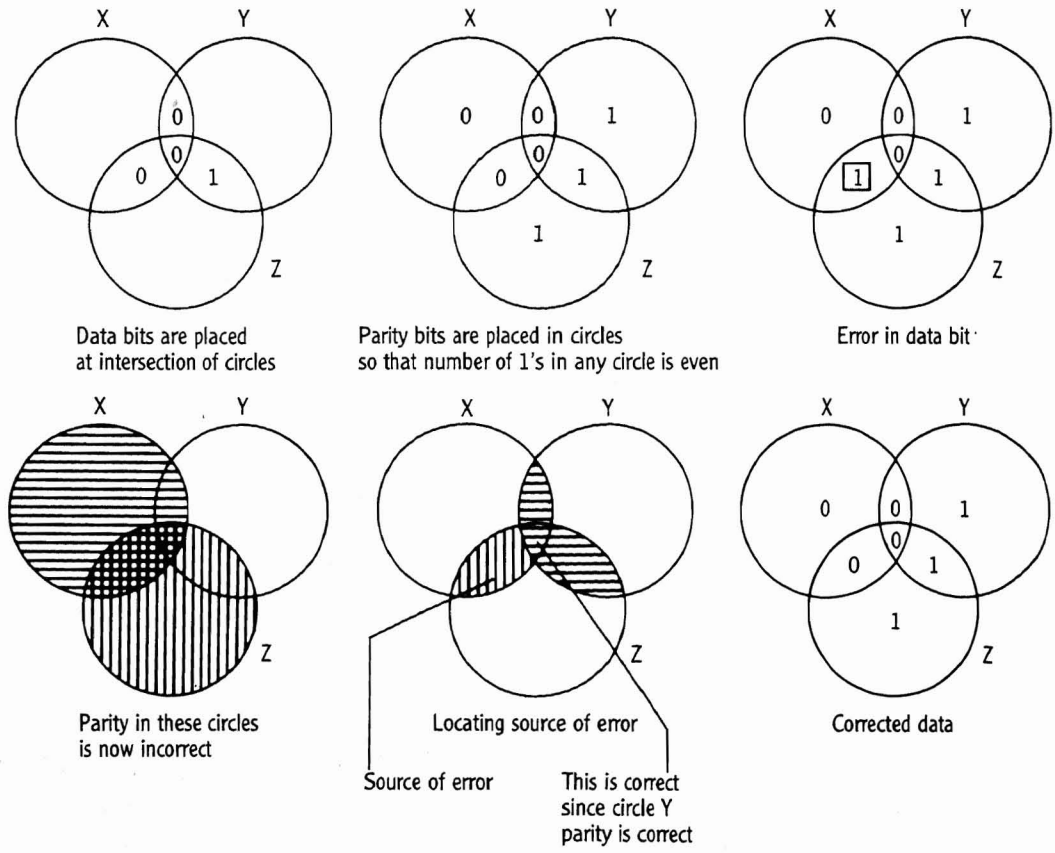
Any received message falling into a shaded sphere is assigned to the single legal message located at the center of the sphere (dot). Messages falling into the unshaded regions cannot be corrected.

The above (and later) schemes developed to enhance computer and communication reliability do not really provide an adequate explanation of how the brain operates in the presence of failure, and they certainly do not explain the ultra-reliable "operation" of whole species or societies of intelligent organisms. In a sense, these "fault tolerant" schemes slow the effects of degeneration; they do not provide a mechanism for compensation, regeneration, or evolutionary improvement.

It has been suggested ([H. Crane, in press] and Box 2-3) that the brain is literally a collection of intelligent agents operating as a tightly knit social system, and that the same dynamics that allows for the malfunction or even death of individuals in society underlies the ability of the brain to function in the presence of cell death and local processing errors.

DISCUSSION

The brain is a mystery we may never succeed in penetrating—in addition to the obvious difficulties of discovering



(a)

0001011	1000000
0001100	1000111
0010010	1011001
0010101	1011110
0100001	1101010
0100110	1101101
0111000	1110011
0111111	1110100

(b)

FIGURE 2-14 Using Parity for Single Error Detection and Correction of a Message.

(a) Venn diagram explanation of a message coding scheme. (b) A set of messages for four data bits and three parity bits. The set of 16 messages shown can be correctly decoded even if a single binary symbol is incorrectly received in any transmitted message. If instead, we use simple replication of a four-bit code for each of the 16 possible messages, then three replications requiring 12 bits must be employed to achieve the same level of error recovery.

the nature of such an amazingly complex system, mathematics teaches us that fallacies and paradoxes are introduced into any analytical framework that is capable of discussing or examining itself. As Hofstadter has said [Hofstadter 79]:

> All the limitative theorems of metamathematics and the theory of computation suggest that once the ability to represent your own structure has reached a certain critical point, that is the kiss of death; it guarantees that you can never represent yourself totally. Gödel's Incompleteness Theorem, Church's Undecidability Theorem, Turing's Halting Theorem, Tarski's Truth Theorem—all have the flavor of some ancient fairy tale which warns you that, *To seek self-knowledge is to embark on a journey which . . . will always be incomplete, cannot be charted on any map, will never halt, cannot be described.* [p.697]

We might then ask how human intelligence can seemingly bypass the barriers imposed by logical proofs of unsolvability and noncomputability—or even those of intrinsically difficult (though solvable) problems. We note that it is often easier to prove the correctness of a result than to find the correct answer in the first place. If an "illogical" system, employing induction and analogy (see Chapter 4), can make a sufficiently high percentage of good guesses and pass them on to a logically correct checking device, the combination may be capable of effective operation even in situations where a logically consistent mechanical system will fail. The parity problem (Box 2-4) is an example where human intuition can easily find an answer, while no mechanical procedure has yet been devised to solve this particular formulation of the problem. However, both logical and nonlogical mechanisms must generally contend with the nonsymmetry of solution versus no-solution; if we can obtain a solution (e.g., by guessing), and demonstrate or prove it, we have solved our problem; but if we cannot find an answer, we can almost never be sure that a solution does not exist.

Finally, an interesting and important question is, "What does it mean to know something?" The scientific viewpoint, grounded in the concept of operationalism, is that to know or understand something is to be able to predict its behavior. We usually express our knowledge of things by building mechanical or symbolic models, and relating the behavior of the model to the situation we wish to understand. It may be obvious that some very complex things (e.g., the universe) may not be understandable by any system less complex than the thing itself (or even have a description of lower complexity). However, it is not intuitively obvious that the specific way we attempt to express a problem, or the way we choose to describe the answer, should radically affect the difficulty of finding a solution. This assumes that we have not altered the competence of the system to deal with the problem, or the amount of information available to the system, but that we merely select a logically equivalent but different "phrasing." A dramatic example of this situation was presented in Box 2-4, in which a change in the way we are permitted to present (represent) the answer to the "parity problem" changes it from an unsolved problem to a trivial one. Finding effective representations appears to be at the heart of intelligent behavior; this is an issue we come back to repeatedly in the remainder of this book.

Appendixes

2-1

The Nerve Cell and Nervous System Organization

Plants do not have specialized cells (nerve cells) to transmit sensory and control information. While some very simple organisms[12] have specialized structures[13] that respond to external stimuli and coordinate movement of cell structures such as cilia, the specialized nerve cell is one of the main distinguishing attributes of members of the animal kingdom. All major groups of multicellular animals except the sponges have definite nerve cells (the sponges employ chemical means for internal coordination).

The nerve net, the most primitive system of organization of nerve cells, is found in the hydra (Fig. 2-15b). When any part of the hydra is stimulated, activity spreads out along the nerve net in all possible directions, eventually involving the entire organism. In addition to the more highly organized "nervous systems" based on one or more nerve cords and nerve cell concentrations called "ganglia," nerve nets are found in the blood vessels and intestinal walls of all vertebrates (including man).

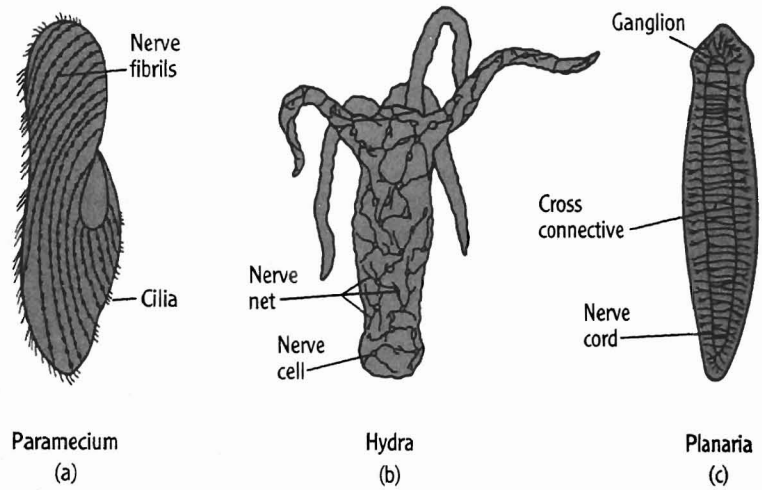


FIGURE 2-15 Nervous Systems of Various Primitive Organisms.

(a) Paramecium. (b) Hydra. (c) Planaria. (From *Biological Science: Molecules to Man*, BSCS Blue Version, 2nd edition, Houghton Mifflin, Boston, 1968, with permission.)

The planarian is one of the simplest organisms with a nervous system in addition to a nerve net (see Fig. 2-15c). A separate nerve cord runs along each side of its body terminating in a ganglion at the head end of the organism. It is quite possible that the nervous systems found in all higher organisms are merely size and complexity elaborations of the basic structure of the nervous system of the planarian.

The Nerve Cell

A nervous system is an organized network of nerve cells or neurons. Between seven and 100 different classes of neurons[14] have been identified in the human nervous system, three of which are shown in Fig. 2-16. Some of these cells are as long

[12]For example, the single cell Paramecium (kingdom Protista).

[13]For example, nerve fibrils as shown in Fig. 2-15a.

[14]Different definitions, based on somewhat arbitrary criteria, have been employed for classifying neurons.

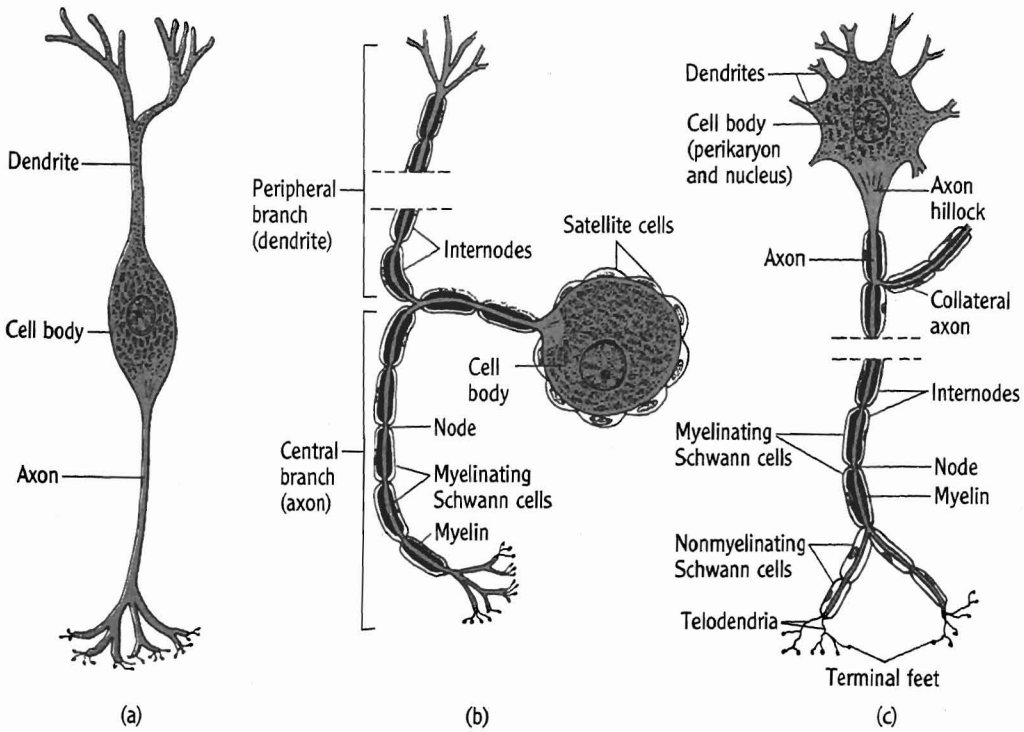


FIGURE 2-16 The Neuron.

Three different types of neurons. Myelin sheaths are shown in black. (a) Bipolar neuron (as found in the retina of the eye). (b) Pseudo-unipolar neuron (myelinated sensory neuron). (c) Multipolar neuron (myelinated somatic motor neuron). (From E.L. Weinreb. *Anatomy and Physiology*. Addison-Wesley, Reading, Mass., 1984, p. 135, with permission.)

as 3 meters, and depending on cell characteristics, nerve impulses travel at rates varying from 10 to 120 meters per second. As shown in Figure 2-16a, the typical nerve cell consists of three parts, the dendrites, the cell body, and the axon (also called the nerve fiber). The dendrites carry nerve signals toward the cell body, while the axon carries signals away. The nucleus of the neuron, located in the cell body, varies in form in different animals, and even within different parts of the nervous system of the same animal.

Nerve structures are formed from bundles of neurons, arranged with the end branches of the axon of one neuron lying close to the dendrites of another neuron (Fig. 2-17). Each nerve cell can directly interact with up to 200,000 other neurons, although a more typical number of interacting neurons is somewhere between 1000 and 10,000. The point of contact between the components of two neurons is called a synapse. A small microscopic gap between the two cells exists at the synapse, and it is known that the ease of nerve signal transmission across the synapse is altered by activity in the nervous system—a possible mechanism for learning.

If the end of a nerve fiber is sufficiently stimulated (i.e., there is a "threshold" below which the nerve cell does not respond) , the stimulus starts chemical and electrical changes that travel over the length of the fiber. These changes are

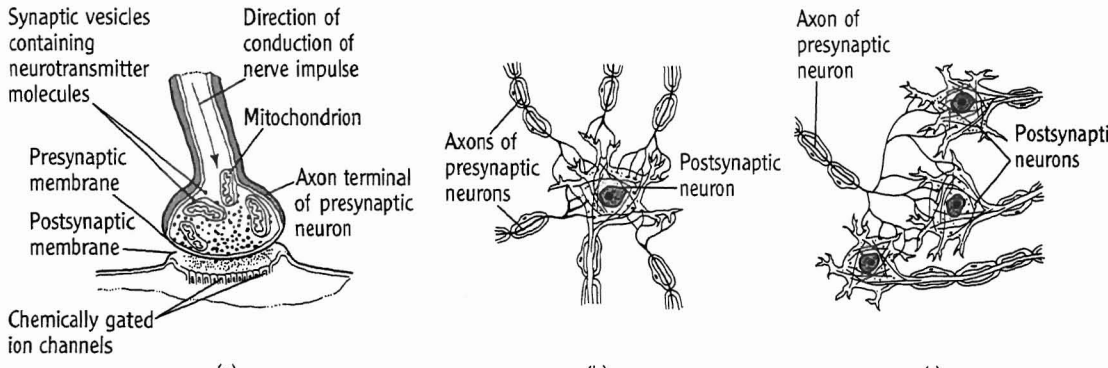


FIGURE 2-17 Nerve Structures.

(a) A chemical synapse. When a nerve impulse arrives at an axon terminal, chemical neurotransmitter molecules are released. The molecules diffuse across the synaptic cleft and attach to receptors on the membrane of the postsynaptic neuron. This attachment alters the three-dimensional shapes of the receptors and initiates a series of events that influence the activity of the postsynaptic neuron. (b) Many neuronal processes converge on a single cell. (c) Neuronal processes of a single cell diverge to a number of other cells. (From A. P. Spence and E. B. Mason. *Human Anatomy and Physiology*, Benjamin Cummings Publishing Co., Menlo Park, California, 1983, with permission.)

called the nerve impulse. After the impulse passes a segment of the nerve fiber, the segment recovers to its original state and is then ready for a new impulse.

Information Coding

One of the primary purposes of the neuron is to convey information. However, the brain uses stereotyped electrical signals for this purpose. These signals are virtually identical in all the nerve cells of all animals; i.e., they are symbols that do not in any way resemble the objects or concepts they represent. Thus, the origins and destinations of the nerve fibers must determine much of the content of the information they transmit. For example, signals reaching the brain from the optic nerve are known to contain visual as opposed to auditory information. In addition to the implicit source information, it is generally assumed that the only other piece of information a neuron can transmit is the equivalent of a single number (e.g., a magnitude representing the strength of a stimulus). Since the neuron nominally has an all-or-none response, it cannot use signal amplitude to encode magnitude information, but instead must use rate of firing or frequency. Neurons have a maximum firing rate of 1000 pulses per second.

Computation

It is generally assumed that the role the neuron plays in the brain's reasoning processes is equivalent to that of a logical switching element in a digital computer. While this is almost certainly too simple an analogy, we note that the neuron can compute a class of logical functions, called "threshold functions," since it has a sensitivity threshold, adjustable (adaptive) signal attenuation at each synapse, and an internal structure which allows the energy of incoming nerve signals to be integrated over both space and time. Thus, signals coming into different synapses at the same time, or even into the same synapse at different times, are "weighted" by the synapses and the resulting quantities summed. If this sum exceeds the sensitivity threshold, the neuron fires. The threshold functions include all the logical functions needed to construct a general purpose digital computer.

Further discussions can be found in Kuffler [Kuffler 76].

APPENDIX 2-2

2-2

The Digital Computer

The first mechanical calculating devices were developed at least as early as the second century B.C. In more recent times, Pascal built a calculating machine in the seventeenth century. In the early 1800s, Joseph Marie Jacquard of France developed the idea of using a punched hole in a card to represent a number and control the operation of a loom. Charles Babbage used the Jacquard concept for his analytical engine in 1833, a machine he worked on until his death in 1871. This machine was quite close in concept to the ideas of the Harvard Mark I, developed almost a century later. Babbage's engine consists of two parts, a "store" to hold all the variables to be operated upon and for preserving previous results, and a "mill" into which the quantities to be operated upon are brought. Two sets of cards are used, one to direct the operations, and the other to hold the values of the variables that are to be operated upon. Augusta Ada Byron, the mathematically trained daughter of Lord Byron, wrote about the analytical engine: "We may say most aptly that the analytical engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves." The mechanical complexities of the device and lack of financial support prevented Babbage from completing his engine.

Toward the end of the century (1886) Herman Hollerith realized that punched holes could be sensed by a machine to sort and manipulate the numbers represented by the holes. Hollerith cards and the associated machines were used for tabulation and statistical analysis by the U.S. Census Bureau. The first digital computer was the Harvard Mark I (automatic sequence controlled calculator, 1939). The operation of the machine was controlled by a plugboard that was wired to obtain a desired computation sequence; the arithmetic operations were carried out using relays. By 1946, the ENIAC, an all-electronic computer using vacuum tubes, replaced the electromechanical computer. It was a thousand times faster than the electromechanical devices, but still used a plugboard for control.

An important conceptual advance came at about the same time, when John von Neumann, Arthur Burks, and Herman Goldstine wrote an influential report, "Preliminary Discussion of an Electronic Computing Instrument." The report proposed a "stored program" concept to replace plugboards and programming switches. The control of the machine was to be carried out by means of a sequence of instruction codes stored as numerals in the memory of the computer. This so-called von Neumann architecture is the basis for the modern computer.

As discussed earlier, a computing device must have some way of storing its instructions and data, a means of manipulating the data, and some way of communicating with the user or the outside world. The memory of the computer stores the data and the instructions (the program) prepared by the user. The arithmetic operations (e.g., addition or subtraction) or logic operations (compare two quantities) are carried out in the arithmetic/logic unit. Communication with the outside world is carried out using an input device, such as a keyboard or a visual sensor, and the output can be printed, displayed on a screen, or used to activate a mechanical effector. The operation of the computer is "orchestrated" by the control portion of the system.

A binary coding scheme is often used to represent numbers and symbols in the computer. The reason for this representation is that there are electrical and magnetic circuits and devices that can be reliably switched into one of two stable states. Thus, a decimal number such as seven, represented in binary notation as 111, would appear in the computer as a sequence of three storage devices in the "1" or "on" state. The arithmetic unit is designed so that when it is given two numbers in binary form, it will carry out the required arithmetic operation and return the result in binary form.

It should be kept in mind that the term "memory" as used in the computer is not meant to indicate the type of capabilities possessed by the human memory. The computer memory can be thought of as consisting of ordered slots, each with an "address" in which data are stored. Data is retrieved by accessing the

contents of the memory at a particular memory address, not by automatically linking data items by meaning. The programmer must devise specific accessing schemes to attain some desired form of data association. Much of the effort in artificial intelligence consists of devising representations that can overcome the address-based organization of the computer memory.

The operation of the computer is controlled by a "program," a set of instructions stored in the computer memory. The program specifies the data to be used and the operations to be carried out on the data. Conceptually, the program will eventually be converted into a set of instructions in which (for each instruction) one or more operands are extracted from computer memory, some simple arithmetic or logical operation performed, and the result returned to some new location in memory. All of the final program specifications are given in the form of binary operation codes that can be interpreted by the machine. Some of the instructions are "conditional " in nature, i.e., the next step to be carried out depends on the results of the computation. For example, a conditional instruction might be: "If the result of the current operation is positive, go to step 31, otherwise go to step 240." (This instruction is, of course, binary-coded and not in English.) The use of conditional instructions gives the programmer the ability to write programs that can react to the intermediate results of the computation; otherwise a program would merely carry out the same fixed sequence of operations regardless of the nature of the data.

The control unit examines the next instruction of the program, determines which of the other computer units will be needed, and sends the necessary control codes to each such unit. The timing of the computer operations is accomplished through the use of a "clock," a circuit that produces a continual sequence of timing pulses that synchronize the operation of the various computer elements.

Because it is very tedious to write programs using the primitive binary code required by the computer, programming languages (e.g., BASIC, Pascal, LISP) have been developed that allow the user to specify the desired operations at a higher conceptual level. These high-level language operations are converted by a "compiler" program into more primitive instructions, and then further translated into the low-level binary code required by the computer using an "assembler" program.[15]

For example, a high-level command such as *Add A to B and assign the result to C* will be converted to operations such as *Assign memory locations to numerical quantities A, B, and C. Retrieve A from memory and place it in register 1, retrieve B from memory and place it in register 2, add register 1 to register 2 and place the results in register 3. then store the contents of register 3 in memory location C.* These detailed instructions are finally converted to the computer's binary code.

[15]The compiler can be independent of the specific computer on which the program is to be run, but the assembler is usually specific to a particular type of computer. The compiler and assembler are often combined into a single program for more efficient operation.