3.0 Chapter Overview

When cognitive science arose in the late 1950s, it did so in the form of what is now known as the classical approach. Inspired by the nature of the digital electronic computer, classical cognitive science adopted the core assumption that cognition was computation. The purpose of the current chapter is to explore the key ideas of classical cognitive science that provide the core elements of this assumption.

The chapter begins by showing that the philosophical roots of classical cognitive science are found in the rationalist perspective of Descartes. While classical cognitive scientists agree with the Cartesian view of the infinite variety of language, they do not use this property to endorse dualism. Instead, taking advantage of modern formal accounts of information processing, they adopt models that use recursive rules to manipulate the components of symbolic expressions. As a result, finite devices—physical symbol systems—permit an infinite behavioural potential. Some of the key properties of physical symbol systems are reviewed.

One consequence of viewing the brain as a physical substrate that brings a universal machine into being is that this means that cognition can be simulated by other universal machines, such as digital computers. As a result, the computer simulation of human cognition becomes a critical methodology of the classical approach. One issue that arises is validating such simulations. The notions of weak
and strong equivalence are reviewed, with the latter serving as the primary goal of classical cognitive science.

To say that two systems—such as a simulation and a human subject—are strongly equivalent is to say that both are solving the same information processing problem, using the same algorithm, based on the same architecture. Establishing strong equivalence requires collecting behavioural evidence of the types introduced in Chapter 2 (relative complexity, intermediate state, and error evidence) to reverse engineer a subject’s algorithm. It also requires discovering the components of a subject’s architecture, which involves behavioural evidence concerning cognitive impenetrability as well as biological evidence about information processing in the brain (e.g., evidence about which areas of the brain might be viewed as being information processing modules). In general, the search for strong equivalence by classical cognitive scientists involves conducting a challenging research program that can be described as functional analysis or reverse engineering.

The reverse engineering in which classical cognitive scientists are engaged involves using a variety of research methods adopted from many different disciplines. This is because this research strategy explores cognition at all four levels of investigation (computational, algorithmic, architectural, and implementational) that were introduced in Chapter 2. The current chapter is organized in a fashion that explores computational issues first, and then proceeds through the remaining levels to end with some considerations about implementational issues of importance to classical cognitive science.

3.1 Mind, Disembodied

In the seventh century, nearly the entire Hellenistic world had been conquered by Islam. The Greek texts of philosophers such as Plato and Aristotle had already been translated into Syriac; the new conquerors translated these texts into Arabic (Kuhn, 1957). Within two centuries, these texts were widely available in educational institutions that ranged from Baghdad to Cordoba and Toledo. By the tenth century, Latin translations of these Arabic texts had made their way to Europe. Islamic civilization “preserved and proliferated records of ancient Greek science for later European scholars” (Kuhn, 1957, p. 102).

The availability of the ancient Greek texts gave rise to scholasticism in Europe during the middle ages. Scholasticism was central to the European universities that arose in the twelfth century, and worked to integrate key ideas of Greek philosophy into the theology of the Church. During the thirteenth century, scholasticism achieved its zenith with the analysis of Aristotle’s philosophy by Albertus Magnus and Thomas Aquinas.
Scholasticism, as a system of education, taught its students the wisdom of the ancients. The scientific revolution that took flight in the sixteenth and seventeenth centuries arose in reaction to this pedagogical tradition. The discoveries of such luminaries as Newton and Leibniz were only possible when the ancient wisdom was directly questioned and challenged.

The seventeenth-century philosophy of René Descartes (1996, 2006) provided another example of fundamental insights that arose from a reaction against scholasticism. Descartes’ goal was to establish a set of incontestable truths from which a rigorous philosophy could be constructed, much as mathematicians used methods of deduction to derive complete geometries from a set of foundational axioms. “The only order which I could follow was that normally employed by geometers, namely to set out all the premises on which a desired proposition depends, before drawing any conclusions about it” (Descartes, 1996, p. 9).

Descartes began his search for truth by applying his own, new method of inquiry. This method employed extreme skepticism: any idea that could possibly be doubted was excluded, including the teachings of the ancients as endorsed by scholasticism. Descartes, more radically, also questioned ideas supplied by the senses because “from time to time I have found that the senses deceive, and it is prudent never to trust completely those who have deceived us even once” (Descartes, 1996, p. 12). Clearly this approach brought a vast number of concepts into question, and removed them as possible foundations of knowledge.

What ideas were removed? All notions of the external world could be false, because knowledge of them is provided by unreliable senses. Also brought into question is the existence of one’s physical body, for the same reason. “I shall consider myself as not having hands or eyes, or flesh, or blood or senses, but as falsely believing that I have all these things” (Descartes, 1996, p. 15).

Descartes initially thought that basic, self-evident truths from mathematics could be spared, facts such as $2 + 3 = 5$. But he then realized that these facts too could be reasonably doubted.

How do I know that God has not brought it about that I too go wrong every time I add two and three or count the sides of a square, or in some even simpler matter, if that is imaginable? (Descartes, 1996, p. 14)

With the exclusion of the external world, the body, and formal claims from mathematics, what was left for Descartes to believe in? He realized that in order to doubt, or even to be deceived by a malicious god, he must exist as a thinking thing. “I must finally conclude that this proposition, I am, I exist, is necessarily true whenever it is put forward by me or conceived in my mind” (Descartes, 1996, p. 17). And what is a thinking thing? “A thing that doubts, understands, affirms, denies, is willing, is unwilling, and alsoimagines and has sensory perceptions” (p. 19).
After establishing his own existence as incontestably true, Descartes used this fact to prove the existence of a perfect God who would not deceive. He then established the existence of an external world that was imperfectly sensed.

However, a fundamental consequence of Descartes’ analysis was a profound division between mind and body. First, Descartes reasoned that mind and body must be composed of different “stuff.” This had to be the case, because one could imagine that the body was divisible (e.g., through losing a limb) but that the mind was impossible to divide.

Indeed the idea I have of the human mind, in so far as it is a thinking thing, which is not extended in length, breadth or height and has no other bodily characteristics, is much more distinct than the idea of any corporeal thing. (Descartes, 1996, p. 37)

Further to this, the mind was literally disembodied—the existence of the mind did not depend upon the existence of the body.

Accordingly this ‘I,’ that is to say, the Soul by which I am what I am, is entirely distinct from the body and is even easier to know than the body; and would not stop being everything it is, even if the body were not to exist. (Descartes, 2006, p. 29)

Though Descartes’ notion of mind was disembodied, he acknowledged that mind and body had to be linked in some way. The interaction between mind and brain was famously housed in the pineal gland: “The mind is not immediately affected by all parts of the body, but only by the brain, or perhaps just by one small part of the brain, namely the part which is said to contain the ‘common’ sense” (Descartes, 1996, p. 59). What was the purpose of this type of interaction? Descartes noted that the powers of the mind could be used to make decisions beneficial to the body, to which the mind is linked: “For the proper purpose of the sensory perceptions given me by nature is simply to inform the mind of what is beneficial or harmful for the composite of which the mind is a part” (p. 57).

For Descartes the mind, as a thinking thing, could apply various rational operations to the information provided by the imperfect senses: sensory information could be doubted, understood, affirmed, or denied; it could also be elaborated via imagination. In short, these operations could not only inform the mind of what would benefit or harm the mind-body composite, but could also be used to plan a course of action to obtain the benefits or avoid the harm. Furthermore, the mind—via its capacity for willing—could cause the body to perform the desired actions to bring this plan into fruition. In Cartesian philosophy, the disembodied mind was responsible for the “thinking” in a sense-think-act cycle that involved the external world and the body to which the mind was linked.

Descartes’ disembodiment of the mind—his claim that the mind is composed of different “stuff” than is the body or the physical world—is a philosophical position called dualism. Dualism has largely been abandoned by modern science, including cognitive science. The vast majority of cognitive scientists adopt a very different
philosophical position called materialism. According to materialism, the mind is caused by the brain. In spite of the fact that it has abandoned Cartesian dualism, most of the core ideas of classical cognitive science are rooted in the ideas that Descartes wrote about in the seventeenth century. Indeed, classical cognitive science can be thought of as a synthesis between Cartesian philosophy and materialism. In classical cognitive science, this synthesis is best expressed as follows: cognition is the product of a physical symbol system (Newell, 1980). The physical symbol system hypothesis is made plausible by the existence of working examples of such devices: modern digital computers.

3.2 Mechanizing the Infinite

We have seen that the disembodied Cartesian mind is the thinking thing that mediates the sensing of, and acting upon, the world. It does so by engaging in such activities as doubting, understanding, affirming, denying, perceiving, imagining, and willing. These activities were viewed by Descartes as being analogous to a geometer’s use of rules to manipulate mathematical expressions. This leads us to ask, in what medium is thought carried out? What formal rules does it employ? What symbolic expressions does it manipulate?

Many other philosophers were sympathetic to the claim that mental activity was some sort of symbol manipulation. Thomas Hobbes is claimed as one of the philosophical fathers of classical cognitive science because of his writings on the nature of the mind:

When a man Reasoneth, hee does nothing else but conceive a summe totall, from Addition of parcels; or conceive a Remainder, from Substraction of one summe from another.” Such operations were not confined to numbers: “These operations are not incident to Numbers only, but to all manner of things that can be added together, and taken one out of another. (Hobbes, 1967, p. 32)

Hobbes noted that geometricians applied such operations to lines and figures, and that logicians applied these operations to words. Thus it is not surprising that Hobbes described thought as mental discourse—thinking, for him, was language-like.

Why were scholars taken by the idea that language was the medium in which thought was conducted? First, they agreed that thought was exceptionally powerful, in the sense that there were no limits to the creation of ideas. In other words, man in principle was capable of an infinite variety of different thoughts. “Reason is a universal instrument which can operate in all sorts of situations” (Descartes, 2006, p. 47). Second, language was a medium in which thought could be expressed, because it too was capable of infinite variety. Descartes expressed this as follows:
For it is a very remarkable fact that there are no men so dull-witted and stupid, not even madmen, that they are incapable of stringing together different words, and composing them into utterances, through which they let their thoughts be known. (Descartes, 2006, p. 47)

Modern linguists describe this as the creative aspect of language (Chomsky, 1965, 1966). “An essential property of language is that it provides the means for expressing indefinitely many thoughts and for reacting appropriately in an indefinite range of new situations” (Chomsky, 1965, p. 6).

While Descartes did not write a great deal about language specifically (Chomsky, 1966), it is clear that he was sympathetic to the notion that language was the medium for thought. This is because he used the creative aspect of language to argue in favor of dualism. Inspired by the automata that were appearing in Europe in his era, Descartes imagined the possibility of having to prove that sophisticated future devices were not human. He anticipated the Turing test (Turing, 1950) by more than three centuries by using language to separate man from machine.

For we can well conceive of a machine made in such a way that it emits words, and even utters them about bodily actions which bring about some corresponding change in its organs . . . but it is not conceivable that it should put these words in different orders to correspond to the meaning of things said in its presence. (Descartes, 2006, p. 46)

Centuries later, similar arguments still appear in philosophy. For instance, why is a phonograph recording of someone's entire life of speech an inadequate simulation of that speech (Fodor, 1968b)? “At the very best, phonographs do what speakers do, not what speakers can do” (p. 129).

Why might it be impossible for a device to do what speakers can do? For Descartes, language-producing machines were inconceivable because machines were physical and therefore finite. Their finite nature made it impossible for them to be infinitely variable.

Although such machines might do many things as well or even better than any of us, they would inevitably fail to do some others, by which we would discover that they did not act consciously, but only because their organs were disposed in a certain way. (Descartes, 2006, pp. 46–47)

In other words, the creativity of thought or language was only possible in the infinite, nonphysical, disembodied mind.

It is this conclusion of Descartes’ that leads to a marked distinction between Cartesian philosophy and classical cognitive science. Classical cognitive science embraces the creative aspect of language. However, it views such creativity from a materialist, not a dualist, perspective. Developments in logic and in computing that have occurred since the seventeenth century have produced a device that
Descartes did not have at his disposal: the physical symbol system. And—seemingly magically—a physical symbol system is a finite artifact that is capable of an infinite variety of behaviour.

By the nineteenth century, the notion of language as a finite system that could be infinitely expressive was well established (Humboldt, 1999, p. 91): “For language is quite peculiarly confronted by an unending and truly boundless domain, the essence of all that can be thought. It must therefore make infinite employment of finite means.” While Humboldt’s theory of language has been argued to presage many of the key properties of modern generative grammars (Chomsky, 1966), it failed to provide a specific answer to the foundational question that it raised: how can a finite system produce the infinite? The answer to that question required advances in logic and mathematics that came after Humboldt, and which in turn were later brought to life by digital computers.

While it had been suspected for centuries that all traditional pure mathematics can be derived from the basic properties of natural numbers, confirmation of this suspicion was only obtained with advances that occurred in the nineteenth and twentieth centuries (Russell, 1993). The “arithmetisation” of mathematics was established in the nineteenth century, in what are called the Dedekind-Peano axioms (Dedekind, 1901; Peano, 1973). This mathematical theory defines three primitive notions: $0$, number, and successor. It also defines five basic propositions: $0$ is a number; the successor of any number is a number; no two numbers have the same successor; $0$ is not the successor of any number; and the principle of mathematical induction. These basic ideas were sufficient to generate the entire theory of natural numbers (Russell, 1993).

Of particular interest to us is the procedure that is used in this system to generate the set of natural numbers. The set begins with $0$. The next number is $1$, which can be defined as the successor of $0$, as $s(0)$. The next number is $2$, which is the successor of $1$, $s(1)$, and is also the successor of the successor of $0$, $s(s(0))$. In other words, the successor function can be used to create the entire set of natural numbers: $0, s(0), s(s(0)), s(s(s(0)))$, and so on.

The definition of natural numbers using the successor function is an example of simple recursion; a function is recursive when it operates by referring to itself. The expression $s(s(0))$ is recursive because the first successor function takes as input another version of itself. Recursion is one method by which a finite system (such as the Dedekind-Peano axioms) can produce infinite variety, as in the set of natural numbers.

Recursion is not limited to the abstract world of mathematics, nor is its only role to generate infinite variety. It can work in the opposite direction, transforming the large and complex into the small and simple. For instance, recursion can be
used to solve a complex problem by reducing it to a simple version of itself. This problem-solving approach is often called divide and conquer (Knuth, 1997).

One example of this is the famous Tower of Hanoi problem (see Figure 3-1), first presented to the world as a wooden puzzle by French mathematician Edouard Lucas in 1883. In this puzzle, there are three locations, A, B, and C. At the start of this problem there is a set of differently sized wooden discs stacked upon one another at location A. Let us number these discs 0, 1, 2, and so on, where the number assigned to a disc indicates its size. The goal for the problem is to move this entire stack to location C, under two restrictions: first, only one disc can be moved at a time; second, a larger disc can never be placed upon a smaller disc.

![Figure 3-1. The starting configuration for a five-disc version of the Tower of Hanoi problem.](image)

The simplest version of the Tower of Hanoi problem starts with only disc 0 at location A. Its solution is completely straightforward: disc 0 is moved directly to location C, and the problem is solved. The problem is only slightly more complicated if it starts with two discs stacked on location A. First, disc 0 is moved to location B. Second, disc 1 is moved to location C. Third, disc 0 is moved from A to C, stacked on top of disc 1, and the problem has been solved.

What about a Tower of Hanoi problem that begins with three discs? To solve this more complicated problem, we can first define a simpler subproblem: stacking discs 0 and 1 on location B. This is accomplished by doing the actions defined in the preceding paragraph, with the exception that the goal location is B for the subproblem. Once this subtask is accomplished, disc 2 can be moved directly to the final goal, location C. Now, we solve the problem by moving discs 0 and 1, which are stacked on B, to location C, by again using a procedure like the one described in the preceding paragraph.

This account of solving a more complex version of the Tower of Hanoi problem points to the recursive nature of divide and conquer: we solve the bigger problem by
first solving a smaller version of the same kind of problem. To move a stack of \( n \) discs to location C, we first move the smaller stack of \( n - 1 \) discs to location B. “Moving the stack” is the same kind of procedure for the \( n \) discs and for the \( n - 1 \) discs. The whole approach is recursive in the sense that to move the big stack, the same procedure must first be used to move the smaller stack on top of the largest disc.

The recursive nature of the solution to the Tower of Hanoi is made obvious if we write a pseudocode algorithm for moving the disks. Let us call our procedure MoveStack (). It will take four arguments: the number of discs in the stack to be moved, the starting location, the “spare” location, and the goal location. So, if we had a stack of three discs at location A, and wanted to move the stack to location C using location B as the spare, we would execute MoveStack (3, A, B, C).

The complete definition of the procedure is as follows:

\[
\text{MoveStack (N, Start, Spare, Goal)}
\]

\[
\text{If N = 0}
\]

\[
\text{Exit}
\]

\[
\text{Else}
\]

\[
\text{MoveStack (N - 1, Start, Goal, Spare)}
\]

\[
\text{MoveStack (1, Start, Spare, Goal)}
\]

\[
\text{MoveStack (N - 1, Spare, Start, Goal)}
\]

\[
\text{EndIf}
\]

Note the explicit recursion in this procedure, because MoveStack () calls itself to move a smaller stack of disks stacked on top of the disk that it is going to move. Note too that the recursive nature of this program means that it is flexible enough to work with any value of \( N \). Figure 3-2 illustrates an intermediate state that occurs when this procedure is applied to a five-disc version of the problem.

---

**Figure 3-2.** An intermediate state that occurs when MoveStack () is applied to a five-disc version of the Tower of Hanoi.
In the code given above, recursion was evident because `MoveStack()` called itself. There are other ways in which recursion can make itself evident. For instance, recursion can produce hierarchical, self-similar structures such as fractals (Mandelbrot, 1983), whose recursive nature is immediately evident through visual inspection. Consider the Sierpinski triangle (Mandelbrot, 1983), which begins as an equilateral triangle (Figure 3-3).

![Figure 3-3. The root of the Sierpinski triangle is an equilateral triangle.](image)

The next step in creating the Sierpinski triangle is to take Figure 3-3 and reduce it to exactly half of its original size. Three of these smaller triangles can be inscribed inside of the original triangle, as is illustrated in Figure 3-4.

![Figure 3-4. The second step of constructing a Sierpinski triangle.](image)

The rule used to create Figure 3-4 can be applied recursively and (in principle) infinitely. One takes the smaller triangle that was used to create Figure 3-4, makes it exactly half of its original size, and inscribes three copies of this still smaller triangle into each of the three triangles that were used to create Figure 3-4. This rule can be
applied recursively to inscribe smaller triangles into any of the triangles that were added to the figure in a previous stage of drawing. Figure 3-5 shows the result when this rule is applied four times to Figure 3-4.

![Figure 3-5. The Sierpinski triangle that results when the recursive rule is applied four times to Figure 3-4.](image)

The Sierpinski triangle, and all other fractals that are created by recursion, are intrinsically self-similar. That is, if one were to take one of the smaller triangles from which Figure 3-4 is constructed and magnify it, one would see still see the hierarchical structure that is illustrated above. The structure of the whole is identical to the (smaller) structure of the parts. In the next section, we see that the recursive nature of human language reveals itself in the same way.

### 3.3 Phrase Markers and Fractals

Consider a finite set of elements (e.g., words, phonemes, morphemes) that can, by applying certain rules, be combined to create a sentence or expression that is finite in length. A language can be defined as the set of all of the possible expressions that can generated in this way from the same set of building blocks and the same set of rules (Chomsky, 1957). From this perspective, one can define a grammar as a device that can distinguish the set of grammatical expressions from all other expressions, including those that are generated from the same elements but which violate the rules that define the language. In modern linguistics, a basic issue to investigate is the nature of the grammar that defines a natural human language.

Chomsky (1957) noted that one characteristic of a natural language such as English is that a sentence can be lengthened by inserting a clause into its midst. As we see in the following section, this means that the grammar of natural languages is complicated enough that simple machines, such as finite state automata, are not powerful enough to serve as grammars for them.
The complex, clausal structure of a natural language is instead captured by a more powerful device—a Turing machine—that can accommodate the regularities of a context-free grammar (e.g., Chomsky, 1957, 1965). A context-free grammar can be described as a set of rewrite rules that convert one symbol into one or more other symbols. The application of these rewrite rules produces a hierarchically organized symbolic structure called a phrase marker (Radford, 1981). A phrase marker is a set of points or labelled nodes that are connected by branches. Nonterminal nodes represent lexical categories; at the bottom of a phrase marker are the terminal nodes that represent lexical categories (e.g., words). A phrase marker for the simple sentence *Dogs bark* is illustrated in Figure 3-6.

![Figure 3-6. A phrase marker for the sentence Dogs bark.](image)

The phrase marker for a sentence can be illustrated as an upside-down tree whose structure is grown from the root node S (for sentence). The application of the rewrite rule $S \rightarrow NP \ VP$ produces the first layer of the Figure 3-6 phrase marker, showing how the nodes NP (noun phrase) and VP (verb phrase) are grown from S. Other rewrite rules that are invoked to create that particular phrase marker are $NP \rightarrow N$, $N \rightarrow N$, $N \rightarrow dogs$, $VP \rightarrow V$, $V \rightarrow V$, and $V \rightarrow bark$. When any of these rewrite rules are applied, the symbol to the left of the → is rewritten as the symbol or symbols to the right. In the phrase marker, this means the symbols on the right of the → are written as nodes below the original symbol, and are connected to the originating node above, as is shown in Figure 3-6.

In a modern grammar called x-bar syntax (Jackendoff, 1977), nodes like NP and VP in Figure 3-6 are symbols that represent phrasal categories, nodes like N and V are symbols that represent lexical categories, and nodes like “and” are symbols that represent categories that are intermediates between lexical categories and phrasal categories. Such intermediate categories are required to capture some regularities in the syntax of natural human languages.

In some instances, the same symbol can be found on both sides of the → in a rewrite rule. For instance, one valid rewrite rule for the intermediate node of a noun
phrase is \( \overline{N} \rightarrow \text{AP} \), where AP represents an adjective phrase. Because the same symbol occurs on each side of the equation, the context-free grammar is recursive. One can apply this rule repeatedly to insert clauses of the same type into a phrase. This is shown in Figure 3-7, which illustrates phrase markers for noun phrases that might apply to my dog Rufus. The basic noun phrase is the dog. If this recursive rule is applied once, it permits a more elaborate noun phrase to be created, as in the cute dog. Recursive application of this rule permits the noun phrase to be elaborated indefinitely, (e.g., the cute brown scruffy dog).

![Phrase markers for three noun phrases: (A) the dog, (B) the cute dog, and (C) the cute brown scruffy dog. Note the recursive nature of (C).](image)

The recursive nature of a context-free grammar is revealed in a visual inspection of a phrase marker like the one illustrated in Figure 3-7C. As one inspects the figure, one sees the same pattern recurring again and again, as was the case with the Sierpinski triangle. The recursive nature of a context-free grammar produces self-similarity
within a phrase marker. The recursion of such a grammar is also responsible for its ability to use finite resources (a finite number of building blocks and a finite number of rewrite rules) to produce a potentially infinite variety of expressions, as in the sentences of a language, each of which is represented by its own phrase marker.

3.4 Behaviourism, Language, and Recursion

Behaviourism viewed language as merely being observable behaviour whose development and elicitation was controlled by external stimuli:

A speaker possesses a verbal repertoire in the sense that responses of various forms appear in his behavior from time to time in relation to identifiable conditions. A repertoire, as a collection of verbal operants, describes the potential behavior of a speaker. To ask where a verbal operant is when a response is not in the course of being emitted is like asking where one's knee-jerk is when the physician is not tapping the patellar tendon. (Skinner, 1957, p. 21)

Skinner’s (1957) treatment of language as verbal behaviour explicitly rejected the Cartesian notion that language expressed ideas or meanings. To Skinner, explanations of language that appealed to such unobservable internal states were necessarily unscientific:

It is the function of an explanatory fiction to allay curiosity and to bring inquiry to an end. The doctrine of ideas has had this effect by appearing to assign important problems of verbal behavior to a psychology of ideas. The problems have then seemed to pass beyond the range of the techniques of the student of language, or to have become too obscure to make further study profitable. (Skinner, 1957, p. 7)

Modern linguistics has explicitly rejected the behaviourist approach, arguing that behaviourism cannot account for the rich regularities that govern language (Chomsky, 1959b).

The composition and production of an utterance is not strictly a matter of stringing together a sequence of responses under the control of outside stimulation and intraverbal association, and that the syntactic organization of an utterance is not something directly represented in any simple way in the physical structure of the utterance itself. (Chomsky, 1959b, p. 55)

Modern linguistics has advanced beyond behaviourist theories of verbal behaviour by adopting a particularly technical form of logicism. Linguists assume that verbal behaviour is the result of sophisticated symbol manipulation: an internal generative grammar.

By a generative grammar I mean simply a system of rules that in some explicit and well-defined way assigns structural descriptions to sentences. Obviously, every
A sentence’s structural description is represented by using a phrase marker, which is a hierarchically organized symbol structure that can be created by a recursive set of rules called a context-free grammar. In a generative grammar another kind of rule, called a transformation, is used to convert one phrase marker into another.

The recursive grammars that have been developed in linguistics serve two purposes. First, they formalize key structural aspects of human languages, such as the embedding of clauses within sentences. Second, they explain how finite resources are capable of producing an infinite variety of potential expressions. This latter accomplishment represents a modern rebuttal to dualism; we have seen that Descartes (1996) used the creative aspect of language to argue for the separate, non-physical existence of the mind. For Descartes, machines were not capable of generating language because of their finite nature.

Interestingly, a present-day version of Descartes’ (1996) analysis of the limitations of machines is available. It recognizes that a number of different information processing devices exist that vary in complexity, and it asks which of these devices are capable of accommodating modern, recursive grammars. The answer to this question provides additional evidence against behaviourist or associationist theories of language (Bever, Fodor, & Garrett, 1968).

![Figure 3-8. How a Turing machine processes its tape.](image)

In Chapter 2, we were introduced to one simple—but very powerful—device, the Turing machine (Figure 3-8). It consists of a machine head that manipulates the symbols on a ticker tape, where the ticker tape is divided into cells, and each cell is capable of holding only one symbol at a time. The machine head can move back and forth along the tape, one cell at a time. As it moves it can read the symbol on the current cell, which can cause the machine head to change its physical state. It is also capable of writing a new symbol on the tape. The behaviour of the machine head—its
new physical state, the direction it moves, the symbol that it writes—is controlled by a machine table that depends only upon the current symbol being read and the current state of the device. One uses a Turing machine by writing a question on its tape, and setting the machine head into action. When the machine head halts, the Turing machine’s answer to the question has been written on the tape.

What is meant by the claim that different information processing devices are available? It means that systems that are different from Turing machines must also exist. One such alternative to a Turing machine is called a finite state automaton (Minsky, 1972; Parkes, 2002), which is illustrated in Figure 3-9. Like a Turing machine, a finite state automaton can be described as a machine head that interacts with a ticker tape. There are two key differences between a finite state machine and a Turing machine.

![Figure 3-9. How a finite state automaton processes the tape. Note the differences between Figures 3-9 and 3-8.](image)

First, a finite state machine can only move in one direction along the tape, again one cell at a time. Second, a finite state machine can only read the symbols on the tape; it does not write new ones. The symbols that it encounters, in combination with the current physical state of the device, determine the new physical state of the device. Again, a question is written on the tape, and the finite state automaton is started. When it reaches the end of the question, the final physical state of the finite state automaton represents its answer to the original question on the tape.

It is obvious that a finite state automaton is a simpler device than a Turing machine, because it cannot change the ticker tape, and because it can only move in one direction along the tape. However, finite state machines are important information processors. Many of the behaviours in behaviour-based robotics are produced using finite state machines (Brooks, 1989, 1999, 2002). It has also been argued that such devices are all that is required to formalize behaviourist or associationist accounts of behaviour (Bever, Fodor, & Garrett., 1968).
What is meant by the claim that an information processing device can “accommodate” a grammar? In the formal analysis of the capabilities of information processors (Gold, 1967), there are two answers to this question. Assume that knowledge of some grammar has been built into a device’s machine head. One could then ask whether the device is capable of accepting a grammar. In this case, the “question” on the tape would be an expression, and the task of the information processor would be to accept the string, if it is grammatical according to the device’s grammar, or to reject the expression, if it does not belong to the grammar. Another question to ask would be whether the information processor is capable of generating the grammar. That is, given a grammatical expression, can the device use its existing grammar to replicate the expression (Wexler & Culicover, 1980)?

In Chapter 2, it was argued that one level of investigation to be conducted by cognitive science was computational. At the computational level of analysis, one uses formal methods to investigate the kinds of information processing problems a device is solving. When one uses formal methods to determine whether some device is capable of accepting or generating some grammar of interest, one is conducting an investigation at the computational level.

One famous example of such a computational analysis was provided by Bever, Fodor, and Garrett (1968). They asked whether a finite state automaton was capable of accepting expressions that were constructed from a particular artificial grammar. Expressions constructed from this grammar were built from only two symbols, a and b. Grammatical strings in the sentence were “mirror images,” because the pattern used to generate expressions was \( b^N a b^N \) where N is the number of bs in the string. Valid expressions generated from this grammar include a, bbbbbabbbb, and bbabb. Expressions that cannot be generated from the grammar include ab, babb, bb, and bbbabb.

While this artificial grammar is very simple, it has one important property: it is recursive. That is, a simple context-free grammar can be defined to generate its potential expressions. This context-free grammar consists of two rules, where Rule 1 is \( S \rightarrow a \), and Rule 2 is \( a \rightarrow bab \). A string is begun by using Rule 1 to generate an a. Rule 2 can then be applied to generate the string bab. If Rule 2 is applied recursively to the central bab then longer expressions will be produced that will always be consistent with the pattern \( b^N a b^N \).

Bever, Fodor, and Garrett (1968) proved that a finite state automaton was not capable of accepting strings generated from this recursive grammar. This is because a finite state machine can only move in one direction along the tape, and cannot write to the tape. If it starts at the first symbol of a string, then it is not capable of keeping track of the number of bs read before the a, and comparing this to the number of bs read after the a. Because it can’t go backwards along the tape, it can’t deal with recursive languages that have embedded clausal structure.
Bever, Fodor, and Garrett (1968) used this result to conclude that associationism (and radical behaviourism) was not powerful enough to deal with the embedded clauses of natural human language. As a result, they argued that associationism should be abandoned as a theory of mind. The impact of this proof is measured by the lengthy responses to this argument by associationist memory researchers (Anderson & Bower, 1973; Paivio, 1986). We return to the implications of this argument when we discuss connectionist cognitive science in Chapter 4.

While finite state automata cannot accept the recursive grammar used by Bever, Fodor, and Garrett (1968), Turing machines can (Révész, 1983). Their ability to move in both directions along the tape provides them with a memory that enables them to match the number of leading bs in a string with the number of trailing bs.

Modern linguistics has concluded that the structure of human language must be described by grammars that are recursive. Finite state automata are not powerful enough devices to accommodate grammars of this nature, but Turing machines are. This suggests that an information processing architecture that is sufficiently rich to explain human cognition must have the same power—must be able to answer the same set of questions—as do Turing machines. This is the essence of the physical symbol system hypothesis (Newell, 1980), which are discussed in more detail below. The Turing machine, as we saw in Chapter 2 and further discuss below, is a universal machine, and classical cognitive science hypothesizes that “this notion of symbol system will prove adequate to all of the symbolic activity this physical universe of ours can exhibit, and in particular all the symbolic activities of the human mind” (Newell, 1980, p. 155).

### 3.5 Underdetermination and Innateness

The ability of a device to accept or generate a grammar is central to another computational level analysis of language (Gold, 1967). Gold performed a formal analysis of language learning which revealed a situation that is known as Gold’s paradox (Pinker, 1979). One solution to this paradox is to adopt a position that is characteristic of classical cognitive science, and which we have seen is consistent with its Cartesian roots. This position is that a good deal of the architecture of cognition is innate.

Gold (1967) was interested in the problem of how a system could learn the grammar of a language on the basis of a finite set of example expressions. He considered two different situations in which the learning system could be presented with expressions. In informant learning, the learner is presented with either valid or invalid expressions, and is also told about their validity, i.e., told whether they belong to the grammar or not. In text learning, the only expressions that are presented to the learner are grammatical.
Whether a learner is undergoing informant learning or text learning, Gold (1967) assumed that learning would proceed as a succession of presentations of expressions. After each expression was presented, the language learner would generate a hypothesized grammar. Gold proposed that each hypothesis could be described as being a Turing machine that would either accept the (hypothesized) grammar or generate it. In this formalization, the notion of “learning a language” has become “selecting a Turing machine that represents a grammar” (Osherson, Stob, & Weinstein, 1986).

According to Gold’s (1967) algorithm, a language learner would have a current hypothesized grammar. When a new expression was presented to the learner, a test would be conducted to see if the current grammar could deal with the new expression. If current grammar succeeded, then it remained. If the current grammar failed, then a new grammar—a new Turing machine—would have to be selected.

Under this formalism, when can we say that a grammar has been learned? Gold defined language learning as the identification of the grammar in the limit. When a language is identified in the limit, this means that the current grammar being hypothesized by the learner does not change even as new expressions are encountered. Furthermore, it is expected that this state will occur after a finite number of expressions have been encountered during learning.

In the previous section, we considered a computational analysis in which different kinds of computing devices were presented with the same grammar. Gold (1967) adopted an alternative approach: he kept the information processing constant—that is, he always studied the algorithm sketched above—but he varied the complexity of the grammar that was being learned, and he varied the conditions under which the grammar was presented, i.e., informant learning versus text learning.

In computer science, a formal description of any class of languages (human or otherwise) relates its complexity to the complexity of a computing device that could generate or accept it (Hopcroft & Ullman, 1979; Révész, 1983). This has resulted in a classification of grammars known as the Chomsky hierarchy (Chomsky, 1959a). In the Chomsky hierarchy, the simplest grammars are regular, and they can be accommodated by finite state automata. The next most complicated are context-free grammars, which can be processed by pushdown automata (a device that is a finite state automaton with a finite internal memory). Next are the context-sensitive grammars, which are the domain of linear bounded automata (i.e., a device like a Turing machine, but with a ticker tape of bounded length). The most complex grammars are the generative grammars, which can only be dealt with by Turing machines.

Gold (1967) used formal methods to determine the conditions under which each class of grammars could be identified in the limit. He was able to show that text learning could only be used to acquire the simplest grammar. In contrast, Gold
found that informant learning permitted context-sensitive and context-free grammars to be identified in the limit.

Gold’s (1967) research was conducted in a relatively obscure field of theoretical computer science. However, Steven Pinker brought it to the attention of cognitive science more than a decade later (Pinker, 1979), where it sparked a great deal of interest and research. This is because Gold’s computational analysis revealed a paradox of particular interest to researchers who studied how human children acquire language.

Gold’s (1967) proofs indicated that informant learning was powerful enough that a complex grammar can be identified in the limit. Such learning was not possible with text learning. Gold’s paradox emerged because research strongly suggests that children are text learners, not informant learners (Pinker, 1979, 1994, 1999). It is estimated that 99.93 percent of the language to which children are exposed is grammatical (Newport, Gleitman, & Gleitman, 1977). Furthermore, whenever feedback about language grammaticality is provided to children, it is not systematic enough to be used to select a grammar (Marcus, 1993).

Gold’s paradox is that while he proved that grammars complex enough to model human language could not be text learned, children learn such grammars—and do so via text learning! How is this possible?

Gold’s paradox is an example of a problem of underdetermination. In a problem of underdetermination, the information available from the environment is not sufficient to support a unique interpretation or inference (Dawson, 1991). For instance, Gold (1967) proved that a finite number of expressions presented during text learning were not sufficient to uniquely determine the grammar from which these expressions were generated, provided that the grammar was more complicated than a regular grammar.

There are many approaches available for solving problems of underdetermination. One that is most characteristic of classical cognitive science is to simplify the learning situation by assuming that some of the to-be-learned information is already present because it is innate. For instance, classical cognitive scientists assume that much of the grammar of a human language is innately available before language learning begins.

The child has an innate theory of potential structural descriptions that is sufficiently rich and fully developed so that he is able to determine, from a real situation in which a signal occurs, which structural descriptions may be appropriate to this signal. (Chomsky, 1965, p. 32)

If the existence of an innate, universal base grammar—a grammar used to create phrase markers—is assumed, then a generative grammar of the type proposed by Chomsky can be identified in the limit (Wexler & Culicover, 1980). This is because learning the language is simplified to the task of learning the set of transformations...
that can be applied to phrase markers. More modern theories of transformational grammars have reduced the number of transformations to one, and have described language learning as the setting of a finite number of parameters that determine grammatical structure (Cook & Newson, 1996). Again, these grammars can be identified in the limit on the basis of very simple input expressions (Lightfoot, 1989). Such proofs are critical to cognitive science and to linguistics, because if a theory of language is to be explanatorily adequate, then it must account for how language is acquired (Chomsky, 1965).

Rationalist philosophers assumed that some human knowledge must be innate. This view was reacted against by empiricist philosophers who viewed experience as the only source of knowledge. For the empiricists, the mind was a tabula rasa, waiting to be written upon by the world. Classical cognitive scientists are comfortable with the notion of innate knowledge, and have used problems of underdetermination to argue against the modern tabula rasa assumed by connectionist cognitive scientists (Pinker, 2002, p. 78): “The connectionists, of course, do not believe in a blank slate, but they do believe in the closest mechanistic equivalent, a general-purpose learning device.” The role of innateness is an issue that separates classical cognitive science from connectionism, and will be encountered again when connectionism is explored in Chapter 4.

3.6 Physical Symbol Systems

Special-purpose logic machines had been developed by philosophers in the late nineteenth century (Buck & Hunka, 1999; Jevons, 1870; Marquand, 1885). However, abstract descriptions of how devices could perform general-purpose symbol manipulation did not arise until the 1930s (Post, 1936; Turing, 1936). The basic properties laid out in these mathematical theories of computation define what is now known as a physical symbol system (Newell, 1980; Newell & Simon, 1976). The concept physical symbol system defines “a broad class of systems that is capable of having and manipulating symbols, yet is also realizable within our physical universe” (Newell, 1980, p. 136).

A physical symbol system operates on a finite set of physical tokens called symbols. These are components of a larger physical entity called a symbol structure or a symbolic expression. It also consists of a set of operators that can create, modify, duplicate, or destroy symbols. Some sort of control is also required to select at any given time some operation to apply. A physical symbol system produces, over time, an evolving or changing collection of expressions. These expressions represent or designate entities in the world (Newell, 1980; Newell & Simon, 1976). As a result, the symbol manipulations performed by such a device permit new meanings to be
derived, in the same way as new knowledge is arrived at in the proofs discovered by logicians and mathematicians (Davis & Hersh, 1981).

The abstract theories that describe physical symbol systems were not developed into working artifacts until nearly the midpoint of the twentieth century. “Our deepest insights into information processing were achieved in the thirties, before modern computers came into being. It is a tribute to the genius of Alan Turing” (Newell & Simon, 1976, p. 117). The first digital computer was the Z3, invented in Germany in 1941 by Konrad Zuse (1993). In the United States, the earliest computers were University of Pennsylvania’s ENIAC (created 1943–1946) and EDVAC (created 1945–1950), Harvard’s MARK I (created 1944), and Princeton’s IAS or von Neumann computer (created 1946–1951) (Burks, 2002; Cohen, 1999). The earliest British computer was University of Manchester’s “Baby,” the small-scale experimental machine (SSEM) that was first activated in June, 1948 (Lavington, 1980).

Although specific details vary from machine to machine, all digital computers share three general characteristics (von Neumann, 1958). First, they have a memory for the storage of symbolic structures. In what is now known as the von Neumann architecture, this is a random access memory (RAM) in which any memory location can be immediately accessed—without having to scroll through other locations, as in a Turing machine—by using the memory’s address. Second, they have a mechanism separate from memory that is responsible for the operations that manipulate stored symbolic structures. Third, they have a controller for determining which operation to perform at any given time. In the von Neumann architecture, the control mechanism imposes serial processing, because only one operation will be performed at a time.

Perhaps the earliest example of serial control is the nineteenth-century punched cards used to govern the patterns in silk that were woven by Joseph Marie Jacquard’s loom (Essinger, 2004). During weaving, at each pass of the loom’s shuttle, holes in a card permitted some thread-controlling rods to be moved. When a rod moved, the thread that it controlled was raised; this caused the thread to be visible in that row of the pattern. A sequence of cards was created by tying cards together end to end. When this “chain” was advanced to the next card, the rods would be altered to create the appropriate appearance for the silk pattern’s next row.

The use of punched cards turned the Jacquard loom into a kind of universal machine: one changed the pattern being produced not by changing the loom, but simply by loading it with a different set of punched cards. Thus not only did Jacquard invent a new loom, but he also invented the idea of using a program to control the actions of a machine. Jacquard’s program was, of course, a sequence of punched cards. Their potential for being applied to computing devices in general was recognized by computer pioneer Charles Babbage, who was inspired by Jacquard’s invention (Essinger, 2004).
By the late 1950s, it became conventional to load the program—then known as the “short code” (von Neumann, 1958)—into memory. This is called memory-stored control; the first modern computer to use this type of control was Manchester’s “Baby” (Lavington, 1980). In Chapter 2 we saw an example of this type of control in the universal Turing machine, whose ticker tape memory holds both the data to be manipulated and the description of a special-purpose Turing machine that will do the manipulating. The universal Turing machine uses the description to permit it to pretend to be the specific machine that is defined on its tape (Hodges, 1983).

In a physical symbol system that employs memory-stored control, internal characteristics will vary over time. However, the time scale of these changes will not be uniform (Newell, 1990). The data that is stored in memory will likely be changed rapidly. However, some stored information—in particular, the short code, or what cognitive scientists would call the virtual machine (Pylyshyn, 1984, 1991), that controls processing would be expected to be more persistent. Memory-stored control in turn chooses which architectural operation to invoke at any given time. In a digital computer, the architecture would not be expected to vary over time at all because it is fixed, that is, literally built into the computing device.

The different characteristics of a physical symbol system provide a direct link back to the multiple levels of investigation that were the topic of Chapter 2. When such a device operates, it is either computing some function or solving some information processing problem. Describing this aspect of the system is the role of a computational analysis. The computation being carried out is controlled by an algorithm: the program stored in memory. Accounting for this aspect of the system is the aim of an algorithmic analysis. Ultimately, a stored program results in the device executing a primitive operation on a symbolic expression stored in memory. Identifying the primitive processes and symbols is the domain of an architectural analysis. Because the device is a physical symbol system, primitive processes and symbols must be physically realized. Detailing the physical nature of these components is the goal of an implementational analysis.

The invention of the digital computer was necessary for the advent of classical cognitive science. First, computers are general symbol manipulators. Their existence demonstrated that finite devices could generate an infinite potential of symbolic behaviour, and thus supported a materialist alternative to Cartesian dualism. Second, the characteristics of computers, and of the abstract theories of computation that led to their development, in turn resulted in the general notion of physical symbol system, and the multiple levels of investigation that such systems require.

The final link in the chain connecting computers to classical cognitive science is the logicist assumption that cognition is a rule-governed symbol manipulation of the sort that a physical symbol system is designed to carry out. This produces the
physical symbol system hypothesis: “the necessary and sufficient condition for a physical system to exhibit general intelligent action is that it be a physical symbol system” (Newell, 1980, p. 170). By necessary, Newell meant that if an artifact exhibits general intelligence, then it must be an instance of a physical symbol system. By sufficient, Newell claimed that any device that is a physical symbol system can be configured to exhibit general intelligent action—that is, he claimed the plausibility of machine intelligence, a position that Descartes denied.

What did Newell (1980) mean by general intelligent action? He meant,

the same scope of intelligence seen in human action: that in real situations behavior appropriate to the ends of the system and adaptive to the demands of the environment can occur, within some physical limits. (Newell, 1980, p. 170)

In other words, human cognition must be the product of a physical symbol system. Thus human cognition must be explained by adopting all of the different levels of investigation that were described in Chapter 2.

3.7 Componentiality, Computability, and Cognition

In 1840, computer pioneer Charles Babbage displayed a portrait of loom inventor Joseph Marie Jacquard for the guests at the famous parties in his home (Essinger, 2004). The small portrait was incredibly detailed. Babbage took great pleasure in the fact that most people who first saw the portrait mistook it to be an engraving. It was instead an intricate fabric woven on a loom of the type that Jacquard himself invented.

The amazing detail of the portrait was the result of its being composed of 24,000 rows of weaving. In a Jacquard loom, punched cards determined which threads would be raised (and therefore visible) for each row in the fabric. Each thread in the loom was attached to a rod; a hole in the punched card permitted a rod to move, raising its thread. The complexity of the Jacquard portrait was produced by using 24,000 punched cards to control the loom.

Though Jacquard’s portrait was impressively complicated, the process used to create it was mechanical, simple, repetitive—and local. With each pass of the loom’s shuttle, weaving a set of threads together into a row, the only function of a punched card was to manipulate rods. In other words, each punched card only controlled small components of the overall pattern. While the entire set of punched cards represented the total pattern to be produced, this total pattern was neither contained in, nor required by, an individual punched card as it manipulated the loom’s rods. The portrait of Jacquard was a global pattern that emerged from a long sequence of simple, local operations on the pattern’s components.
In the Jacquard loom, punched cards control processes that operate on local components of the “expression” being weaved. The same is true of the physical symbol systems. Physical symbol systems are finite devices that are capable of producing an infinite variety of potential behaviour. This is possible because the operations of a physical symbol system are recursive. However, this explanation is not complete. In addition, the rules of a physical symbol system are local or componental, in the sense that they act on local components of an expression, not on the expression as a whole.

For instance, one definition of a language is the set of all of its grammatical expressions (Chomsky, 1957). Given this definition, it is logically possible to treat each expression in the set as an unanalyzed whole to which some operation could be applied. This is one way to interpret a behaviourist theory of language (Skinner, 1957): each expression in the set is a holistic verbal behaviour whose likelihood of being produced is a result of reinforcement and stimulus control of the expression as a whole.

However, physical symbol systems do not treat expressions as unanalyzed wholes. Instead, the recursive rules of a physical symbol system are sensitive to the atomic symbols from which expressions are composed. We saw this previously in the example of context-free grammars that were used to construct the phrase markers of Figures 3-6 and 3-7. The rules in such grammars do not process whole phrase markers, but instead operate on the different components (e.g., nodes like S, N, VP) from which a complete phrase marker is constructed.

The advantage of operating on symbolic components, and not on whole expressions, is that one can use a sequence of very basic operations—writing, changing, erasing, or copying a symbol—to create an overall effect of far greater scope than might be expected. As Henry Ford said, nothing is particularly hard if you divide it into small jobs. We saw the importance of this in Chapter 2 when we discussed Leibniz’ mill (Leibniz, 1902), the Chinese room (Searle, 1980), and the discharging of homunculi (Dennett, 1978). In a materialist account of cognition, thought is produced by a set of apparently simple, mindless, unintelligent actions—the primitives that make up the architecture.

The small jobs carried out by a physical symbol system reveal that such a system has a dual nature (Haugeland, 1985). On the one hand, symbol manipulations are purely syntactic—they depend upon identifying a symbol’s type, and not upon semantically interpreting what the symbol stands for. On the other hand, a physical symbol system’s manipulations are semantic—symbol manipulations preserve meanings, and can be used to derive new, sensible interpretations.

Interpreted formal tokens lead two lives: syntactical lives, in which they are meaningless markers, moved according to the rules of some self-contained game; and
Let us briefly consider these two lives. First, we have noted that the rules of a physical symbol system operate on symbolic components of a whole expression. For this to occur, all that is required is that a rule identifies a particular physical entity as being a token or symbol of a particular type. If the symbol is of the right type, then the rule can act upon it in some prescribed way.

For example, imagine a computer program that is playing chess. For this program, the “whole expression” is the total arrangement of game pieces on the chess board at any given time. The program analyzes this expression into its components: individual tokens on individual squares of the board. The physical characteristics of each component token can then be used to identify to what symbol class it belongs: queen, knight, bishop, and so on. Once a token has been classified in this way, appropriate operations can be applied to it. If a game piece has been identified as being a “knight,” then only knight moves can be applied to it—the operations that would move the piece like a bishop cannot be applied, because the token has not been identified as being of the type “bishop.”

Similar syntactic operations are at the heart of a computing device like a Turing machine. When the machine head reads a cell on the ticker tape (another example of componentiality!), it uses the physical markings on the tape to determine that the cell holds a symbol of a particular type. This identification—in conjunction with the current physical state of the machine head—is sufficient to determine which instruction to execute.

To summarize, physical symbol systems are syntactic in the sense that their rules are applied to symbols that have been identified as being of a particular type on the basis of their physical shape or form. Because the shape or form of symbols is all that matters for the operations to be successfully carried out, it is natural to call such systems formal. Formal operations are sensitive to the shape or form of individual symbols, and are not sensitive to the semantic content associated with the symbols.

However, it is still the case that formal systems can produce meaningful expressions. The punched cards of a Jacquard loom only manipulate the positions of thread-controlling rods. Yet these operations can produce an intricate woven pattern such as Jacquard’s portrait. The machine head of a Turing machine reads and writes individual symbols on a ticker tape. Yet these operations permit this device to provide answers to any computable question. How is it possible for formal systems to preserve or create semantic content?

In order for the operations of a physical symbol system to be meaningful, two properties must be true. First, the symbolic structures operated on must have semantic content. That is, the expressions being manipulated must have some relationship to states of the external world that permits the expressions to represent
these states. This relationship is a basic property of a physical symbol system, and is called designation (Newell, 1980; Newell & Simon, 1976). “An expression designates an object if, given the expression, the system can either affect the object itself or behave in ways dependent on the object” (Newell & Simon, 1976, p. 116).

Explaining designation is a controversial issue in cognitive science and philosophy. There are many different proposals for how designation, which is also called the problem of representation (Cummins, 1989) or the symbol grounding problem (Harnad, 1990), occurs. The physical symbol system hypothesis does not propose a solution, but necessarily assumes that such a solution exists. This assumption is plausible to the extent that computers serve as existence proofs that designation is possible.

The second semantic property of a physical symbol system is that not only are individual expressions meaningful (via designation), but the evolution of expressions—the rule-governed transition from one expression to another—is also meaningful. That is, when some operation modifies an expression, this modification is not only syntactically correct, but it will also make sense semantically. As rules modify symbolic structures, they preserve meanings in the domain that the symbolic structures designate, even though the rules themselves are purely formal. The application of a rule should not produce an expression that is meaningless. This leads to what is known as the formalist’s motto: “If you take care of the syntax, then the semantics will take care of itself” (Haugeland, 1985, p. 106).

The assumption that applying a physical symbol system’s rules preserves meaning is a natural consequence of classical cognitive science’s commitment to logicism. According to logicism, thinking is analogous to using formal methods to derive a proof, as is done in logic or mathematics. In these formal systems, when one applies rules of the system to true expressions (e.g., the axioms of a system of mathematics which by definition are assumed to be true [Davis & Hersh, 1981]), the resulting expressions must also be true. An expression’s truth is a critical component of its semantic content.

It is necessary, then, for the operations of a formal system to be defined in such a way that 1) they only detect the form of component symbols, and 2) they are constrained in such a way that manipulations of expressions are meaningful (e.g., truth preserving). This results in classical cognitive science’s interest in universal machines.

A universal machine is a device that is maximally flexible in two senses (Newell, 1980). First, its behaviour is responsive to its inputs; a change in inputs will be capable of producing a change in behaviour. Second, a universal machine must be able compute the widest variety of input-output functions that is possible. This “widest variety” is known as the set of computable functions.
A device that can compute every possible input-output function does not exist. The Turing machine was invented and used to prove that there exist some functions that are not computable (Turing, 1936). However, the subset of functions that are computable is large and important:

It can be proved mathematically that there are infinitely more functions than programs. Therefore, for most functions there is no corresponding program that can compute them. . . . Fortunately, almost all these noncomputable functions are useless, and virtually all the functions we might want to compute are computable. (Hillis, 1998, p. 71)

A major discovery of the twentieth century was that a number of seemingly different symbol manipulators were all identical in the sense that they all could compute the same maximal class of input-output pairings (i.e., the computable functions). Because of this discovery, these different proposals are all grouped together into the class “universal machine,” which is sometimes called the “effectively computable procedures.” This class is “a large zoo of different formulations” that includes “Turing machines, recursive functions, Post canonical systems, Markov algorithms, all varieties of general purpose digital computers, [and] most programming languages” (Newell, 1980, p. 150).

Newell (1980) proved that a generic physical symbol system was also a universal machine. This proof, coupled with the physical symbol system hypothesis, leads to a general assumption in classical cognitive science: cognition is computation, the brain implements a universal machine, and the products of human cognition belong to the class of computable functions.

The claim that human cognition is produced by a physical symbol system is a scientific hypothesis. Evaluating the validity of this hypothesis requires fleshing out many additional details. What is the organization of the program that defines the physical symbol system for cognition (Newell & Simon, 1972)? In particular, what kinds of symbols and expressions are being manipulated? What primitive operations are responsible for performing symbol manipulation? How are these operations controlled? Classical cognitive science is in the business of fleshing out these details, being guided at all times by the physical symbol system hypothesis.

3.8 The Intentional Stance

According to the formalist’s motto (Haugeland, 1985) by taking care of the syntax, one also takes care of the semantics. The reason for this is that, like the rules in a logical system, the syntactic operations of a physical symbol system are constrained to preserve meaning. The symbolic expressions that a physical symbol system evolves will have interpretable designations.
We have seen that the structures a physical symbol system manipulates have two different lives, syntactic and semantic. Because of this, there is a corollary to the formalist's motto, which might be called the semanticist's motto: “If you understand the semantics, then you can take the syntax for granted.” That is, if you have a semantic interpretation of a physical symbol system's symbolic expressions, then you can use this semantic interpretation to predict the future behaviour of the system—the future meanings that it will generate—without having to say anything about the underlying physical mechanisms that work to preserve the semantics.

We have seen that one of the fundamental properties of a physical symbol system is designation, which is a relation between the system and the world that provides interpretations to its symbolic expressions (Newell, 1980; Newell & Simon, 1976). More generally, it could be said that symbolic expressions are intentional—they are about some state of affairs in the world. This notion of intentionality is rooted in the philosophy of Franz Brentano (Brentano, 1995). Brentano used intentionality to distinguish the mental from the physical: “We found that the intentional in-existence, the reference to something as an object, is a distinguishing characteristic of all mental phenomena. No physical phenomenon exhibits anything similar” (p. 97).

To assume that human cognition is the product of a physical symbol system is to also assume that mental states are intentional in Brentano's sense. In accord with the semanticist's motto, the intentionality of mental states can be used to generate a theory of other people, a theory that can be used to predict the behaviour of another person. This is accomplished by adopting what is known as the intentional stance (Dennett, 1987).

The intentional stance uses the presumed contents of someone's mental states to predict their behaviour. It begins by assuming that another person possesses intentional mental states such as beliefs, desires, or goals. As a result, the intentional stance involves describing other people with propositional attitudes.

A propositional attitude is a statement that relates a person to a proposition or statement of fact. For example, if I said to someone “Charles Ives’ music anticipated minimalism,” they could describe me with the propositional attitude “Dawson believes that Charles Ives’ music anticipated minimalism.” Propositional attitudes are of interest to philosophy because they raise a number of interesting logical problems. For example, the propositional attitude describing me could be true, but at the same time its propositional component could be false (for instance, if Ives' music bore no relationship to minimalism at all!). Propositional attitudes are found everywhere in our language, suggesting that a key element of our understanding of others is the use of the intentional stance.

In addition to describing other people with propositional attitudes, the intentional stance requires that other people are assumed to be rational. To assume that a person is rational is to assume that there are meaningful relationships between the
contents of mental states and behaviour. To actually use the contents of mental states to predict behaviour—assuming rationality—is to adopt the intentional stance.

For instance, given the propositional attitudes “Dawson believes that Charles Ives’ music anticipated minimalism” and “Dawson desires to only listen to early minimalist music,” and assuming that Dawson’s behaviour rationally follows from the contents of his intentional states, one might predict that “Dawson often listens to Ives’ compositions.” The assumption of rationality, “in combination with home truths about our needs, capacities and typical circumstances, generates both an intentional interpretation of us as believers and desirers and actual predictions of behavior in great profusion” (Dennett, 1987, p. 50).

Adopting the intentional stance is also known as employing commonsense psychology or folk psychology. The status of folk psychology, and of its relation to cognitive science, provides a source of continual controversy (Christensen & Turner, 1993; Churchland, 1988; Fletcher, 1995; Greenwood, 1991; Haselager, 1997; Ratcliffe, 2007; Stich, 1983). Is folk psychology truly predictive? If so, should the theories of cognitive science involve lawful operations on propositional attitudes? If not, should folk psychology be expunged from cognitive science? Positions on these issues range from eliminative materialism’s argument to erase folk-psychological terms from cognitive science (Churchland, 1988), to experimental philosophy’s position that folk concepts are valid and informative, and therefore should be empirically examined to supplant philosophical concepts that have been developed from a purely theoretical or analytic tradition (French & Wettstein, 2007; Knobe & Nichols, 2008).

In form, at least, the intentional stance or folk psychology has the appearance of a scientific theory. The intentional stance involves using a set of general, abstract laws (e.g., the principle of rationality) to predict future events. This brings it into contact with an important view of cognitive development known as the theory-theory (Gopnik & Meltzoff, 1997; Gopnik, Meltzoff, & Kuhl, 1999; Gopnik & Wellman, 1992; Wellman, 1990). According to the theory-theory, children come to understand the world by adopting and modifying theories about its regularities. That is, the child develops intuitive, representational theories in a fashion that is analogous to a scientist using observations to construct a scientific theory. One of the theories that a child develops is a theory of mind that begins to emerge when a child is three years old (Wellman, 1990).

The scientific structure of the intentional stance should be of no surprise, because this is another example of the logicism that serves as one of the foundations of classical cognitive science. If cognition really is the product of a physical symbol system, if intelligence really does emerge from the manipulation of intentional representations according to the rules of some mental logic, then the semanticist’s
motto should hold. A principle of rationality, operating on propositional attitudes, should offer real predictive power.

However, the logicism underlying the intentional stance leads to a serious problem for classical cognitive science. This is because a wealth of experiments has shown that human reasoners deviate from principles of logic or rationality (Hastie, 2001; Tversky & Kahneman, 1974; Wason, 1966; Wason & Johnson-Laird, 1972). “A purely formal, or syntactic, approach to [reasoning] may suffer from severe limitations” (Wason & Johnson-Laird, 1972, p. 244). This offers a severe challenge to classical cognitive science’s adherence to logicism: if thinking is employing mental logic, then how is it possible for thinkers to be illogical?

It is not surprising that many attempts have been made to preserve logicism by providing principled accounts of deviations from rationalism. Some of these attempts have occurred at the computational level and have involved modifying the definition of rationality by adopting a different theory about the nature of mental logic. Such attempts include rational analysis (Chater & Oaksford, 1999) and probabilistic theories (Oaksford & Chater, 1998, 2001). Other, not unrelated approaches involve assuming that ideal mental logics are constrained by algorithmic and architectural-level realities, such as limited memory and real time constraints. The notion of bounded rationality is a prototypical example of this notion (Chase, Hertwig, & Gigerenzer, 1998; Evans, 2003; Hastie, 2001; Rubinstein, 1998; Simon, Egidi, & Marris, 1995).

The attempts to preserve logicism reflect the importance of the intentional stance, and the semanticist’s motto, to cognitive science. Classical cognitive science is committed to the importance of a cognitive vocabulary, a vocabulary that invokes the contents of mental states (Pylyshyn, 1984).

3.9 Structure and Process

The physical symbol systems of classical cognitive science make a sharp distinction between symbols and the rules that manipulate them. This is called the structure/process distinction. For instance, in a Turing machine the symbols reside in one medium (the ticker tape) that is separate from another medium (the machine head) that houses the operators for manipulating symbols. Whatever the specific nature of cognition’s universal machine, if it is a classical physical symbol system, then it will exhibit the structure/process distinction.

In general, what can be said about the symbols that define the structure that is manipulated by a physical symbol system? It has been argued that cognitive science’s notion of symbol is ill defined (Searle, 1992). Perhaps this is because apart from the need that symbols be physically distinctive, so that they can be identified
as being tokens of a particular type, symbols do not have definitive properties. Symbols are arbitrary, in the sense that anything can serve as a symbol.

The arbitrary nature of symbols is another example of the property of multiple realization that was discussed in Chapter 2.

What we had no right to expect is the immense variety of physical ways to realize any fixed symbol system. What the generations of digital technology have demonstrated is that an indefinitely wide array of physical phenomena can be used to develop a digital technology to produce a logical level of essentially identical character. (Newell, 1980, p. 174)

This is why universal machines can be built out of gears (Swade, 1993), LEGO (Agulló et al., 2003), electric train sets (Stewart, 1994), hydraulic valves, or silicon chips (Hillis, 1998).

The arbitrariness of symbols, and the multiple realization of universal machines, is rooted in the relative notion of universal machine. By definition, a machine is universal if it can simulate any other universal machine (Newell, 1980). Indeed, this is the basic idea that justifies the use of computer simulations to investigate cognitive and neural functioning (Dutton & Starbuck, 1971; Gluck & Myers, 2001; Lewandowsky, 1993; Newell & Simon, 1961; O’Reilly & Munakata, 2000).

For any class of machines, defined by some way of describing its operational structure, a machine of that class is defined to be universal if it can behave like any machine of the class. This puts simulation at the center of the stage. (Newell, 1980, p. 149)

If a universal machine can be simulated by any other, and if cognition is the product of a universal machine, then why should we be concerned about the specific details of the information processing architecture for cognition? The reason for this concern is that the internal aspects of an architecture—the relations between a particular structure-process pairing—are not arbitrary. The nature of a particular structure is such that it permits some, but not all, processes to be easily applied. Therefore some input-output functions will be easier to compute than others because of the relationship between structure and process. Newell and Simon (1972, p. 803) called these second-order effects.

Consider, for example, one kind of representation: a table of numbers, such as Table 3-1, which provides the distances in kilometres between pairs of cities in Alberta (Dawson, Boechler, & Valsangkar-Smyth, 2000). One operation that can easily be applied to symbols that are organized in such a fashion is table lookup. For instance, perhaps I was interested in knowing the distance that I would travel if I drove from Edmonton to Fort McMurray. Applying table lookup to Table 3-1, by looking for the number at the intersection between the Edmonton row and the Fort McMurray column, quickly informs me that the distance is 439 kilometres.
This is because the tabular form of this information makes distances between places explicit, so that they can be “read off of” the representation in a seemingly effortless manner.

Other information cannot be so easily gleaned from a tabular representation. For instance, perhaps I am interested in determining the compass direction that points from Edmonton to Fort McMurray. The table does not make this information explicit—directions between cities cannot be simply read off of Table 3-1.

<table>
<thead>
<tr>
<th></th>
<th>BAN</th>
<th>CAL</th>
<th>CAM</th>
<th>DRU</th>
<th>EDM</th>
<th>FMC</th>
<th>GRA</th>
<th>JAS</th>
<th>LET</th>
<th>LLO</th>
<th>MED</th>
<th>RED</th>
<th>SLA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BANFF</strong></td>
<td></td>
<td>0</td>
<td>128</td>
<td>381</td>
<td>263</td>
<td>401</td>
<td>840</td>
<td>682</td>
<td>287</td>
<td>342</td>
<td>626</td>
<td>419</td>
<td>253</td>
</tr>
<tr>
<td><strong>CALGARY</strong></td>
<td>128</td>
<td>0</td>
<td>274</td>
<td>138</td>
<td>294</td>
<td>733</td>
<td>720</td>
<td>412</td>
<td>216</td>
<td>519</td>
<td>293</td>
<td>145</td>
<td>545</td>
</tr>
<tr>
<td><strong>CAMEO</strong></td>
<td>381</td>
<td>274</td>
<td>0</td>
<td>182</td>
<td>97</td>
<td>521</td>
<td>553</td>
<td>463</td>
<td>453</td>
<td>435</td>
<td>429</td>
<td>129</td>
<td>348</td>
</tr>
<tr>
<td><strong>DRUMHILLER</strong></td>
<td>263</td>
<td>138</td>
<td>182</td>
<td>0</td>
<td>279</td>
<td>703</td>
<td>735</td>
<td>547</td>
<td>282</td>
<td>416</td>
<td>247</td>
<td>165</td>
<td>530</td>
</tr>
<tr>
<td><strong>EDMONTON</strong></td>
<td>401</td>
<td>294</td>
<td>97</td>
<td>279</td>
<td>0</td>
<td>439</td>
<td>456</td>
<td>366</td>
<td>509</td>
<td>251</td>
<td>526</td>
<td>148</td>
<td>250</td>
</tr>
<tr>
<td><strong>FORT MCMURRAY</strong></td>
<td>840</td>
<td>733</td>
<td>521</td>
<td>703</td>
<td>439</td>
<td>0</td>
<td>752</td>
<td>796</td>
<td>948</td>
<td>587</td>
<td>931</td>
<td>587</td>
<td>436</td>
</tr>
<tr>
<td><strong>GRANDE PRAIRIE</strong></td>
<td>682</td>
<td>720</td>
<td>553</td>
<td>735</td>
<td>456</td>
<td>752</td>
<td>0</td>
<td>397</td>
<td>935</td>
<td>701</td>
<td>982</td>
<td>586</td>
<td>318</td>
</tr>
<tr>
<td><strong>JASPER</strong></td>
<td>287</td>
<td>412</td>
<td>463</td>
<td>547</td>
<td>366</td>
<td>796</td>
<td>397</td>
<td>0</td>
<td>626</td>
<td>613</td>
<td>703</td>
<td>413</td>
<td>464</td>
</tr>
<tr>
<td><strong>LETHBRIDGE</strong></td>
<td>342</td>
<td>216</td>
<td>453</td>
<td>282</td>
<td>509</td>
<td>948</td>
<td>935</td>
<td>626</td>
<td>0</td>
<td>605</td>
<td>168</td>
<td>360</td>
<td>760</td>
</tr>
<tr>
<td><strong>LLOYDMINSTER</strong></td>
<td>626</td>
<td>519</td>
<td>245</td>
<td>416</td>
<td>251</td>
<td>587</td>
<td>701</td>
<td>613</td>
<td>605</td>
<td>0</td>
<td>480</td>
<td>374</td>
<td>496</td>
</tr>
<tr>
<td><strong>MEDICINE HAT</strong></td>
<td>419</td>
<td>293</td>
<td>429</td>
<td>247</td>
<td>526</td>
<td>931</td>
<td>982</td>
<td>703</td>
<td>168</td>
<td>480</td>
<td>0</td>
<td>409</td>
<td>777</td>
</tr>
<tr>
<td><strong>RED DEER</strong></td>
<td>253</td>
<td>145</td>
<td>129</td>
<td>165</td>
<td>148</td>
<td>587</td>
<td>586</td>
<td>413</td>
<td>360</td>
<td>374</td>
<td>409</td>
<td>0</td>
<td>399</td>
</tr>
<tr>
<td><strong>SLAVE LAKE</strong></td>
<td>652</td>
<td>545</td>
<td>348</td>
<td>530</td>
<td>250</td>
<td>436</td>
<td>318</td>
<td>464</td>
<td>760</td>
<td>496</td>
<td>777</td>
<td>399</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 3-1.** Distances in kilometres between cities in Alberta, Canada.

However, this does not mean that the table does not contain information about direction. Distance-like data of the sort provided by Table 3-1 can be used as input to a form of factor analysis called multidimensional scaling (MDS) (Romney, Shepard, & Nerlove, 1972; Shepard, Romney, & Nerlove, 1972). This statistical analysis converts the table of distances into a map-like representation of objects that would produce the set of distances in the table. Dawson et al. (2000) performed such an analysis on the Table 3-1 data and obtained the map that is given in Figure 3-10. This map makes the relative spatial locations of the cities obvious; it could be used to simply “read off” compass directions between pairs of places.
“Reading off” information from a representation intuitively means accessing this information easily—by using a small number of primitive operations. If this is not possible, then information might be still be accessed by applying a larger number of operations, but this will take more time. The ease of accessing information is a result of the relationship between structure and process.

The structure-process relationship, producing second-order effects, underscores the value of using relative complexity evidence, a notion that was introduced in Chapter 2. Imagine that a physical symbol system uses a tabular representation of distances. Then we would expect it to compute functions involving distance very quickly, but it would be much slower to answer questions about direction. In contrast, if the device uses a map-like representation, then we would expect it to answer questions about direction quickly, but take longer to answer questions about distance (because, for instance, measuring operations would have to be invoked).

In summary, while structures are arbitrary, structure-process relations are not. They produce second-order regularities that can affect such measures as relative complexity evidence. Using such measures to investigate structure-process relations provides key information about a system's algorithms and architecture.

**Figure 3-10.** Results of applying MDS to Table 3-1.
3.10 A Classical Architecture for Cognition

The physical symbol system hypothesis defines classical cognitive science. This school of thought can be thought of as the modern derivative of Cartesian philosophy. It views cognition as computation, where computation is the rule-governed manipulation of symbols. Thus thinking and reasoning are viewed as the result of performing something akin to logical or mathematical inference. A great deal of this computational apparatus must be innate.

However, classical cognitive science crucially departs from Cartesian philosophy by abandoning dualism. Classical cognitive science instead adopts a materialist position that mechanizes the mind. The technical notion of computation is the application of a finite set of recursive rules to a finite set of primitives to evolve a set of finite symbolic structures or expressions. This technical definition of computation is beyond the capabilities of some devices, such as finite state automata, but can be accomplished by universal machines such as Turing machines or electronic computers. The claim that cognition is the product of a device that belongs to the same class of artifacts such as Turing machines or digital computers is the essence of the physical symbol system hypothesis, and the foundation of classical cognitive science.

Since the invention of the digital computer, scholars have seriously considered the possibility that the brain was also a computer of this type. For instance, the all-or-none nature of a neuron's action potential has suggested that the brain is also digital in nature (von Neumann, 1958). However, von Neumann went on to claim that the small size and slow speed of neurons, in comparison to electronic components, suggested that the brain would have a different architecture than an electronic computer. For instance, von Neumann speculated that the brain's architecture would be far more parallel in nature.

Von Neumann's (1958) speculations raise another key issue. While classical cognitive scientists are confident that brains belong to the same class as Turing machines and digital computers (i.e., all are physical symbol systems), they do not expect the brain to have the same architecture. If the brain is a physical symbol system, then what might its architecture be like?

Many classical cognitive scientists believe that the architecture of cognition is some kind of production system. The model of production system architecture was invented by Newell and Simon (Newell, 1973; Newell & Simon, 1961, 1972) and has been used to simulate many psychological phenomena (Anderson, 1983; Anderson et al., 2004; Anderson & Matessa, 1997; Meyer et al., 2001; Meyer & Kieras, 1997a, 1997b; Newell, 1990; Newell & Simon, 1972). Production systems have a number of interesting properties, including an interesting mix of parallel and serial processing.
A production system is a general-purpose symbol manipulator (Anderson, 1983; Newell, 1973; Newell & Simon, 1972). Like other physical symbol systems, production systems exhibit a marked distinction between symbolic expressions and the rules for manipulating them. They include a working memory that is used to store one or more symbolic structures, where a symbolic structure is an expression that is created by combining a set of atomic symbols. In some production systems (e.g., Anderson, 1983) a long-term memory, which also stores expressions, is present as well. The working memory of a production system is analogous to the ticker tape of a Turing machine or to the random access memory of a von Neumann computer.

The process component of a production system is a finite set of symbol-manipulating rules that are called productions. Each production is a single rule that pairs a triggering condition with a resulting action. A production works by scanning the expressions in working memory for a pattern that matches its condition. If such a match is found, then the production takes control of the memory and performs its action. A production's action is some sort of symbol manipulation—adding, deleting, copying, or moving symbols or expressions in the working memory.

A typical production system is a parallel processor in the sense that all of its productions search working memory simultaneously for their triggering patterns. However, it is a serial processor—like a Turing machine or a digital computer—when actions are performed to manipulate the expressions in working memory. This is because in most production systems only one production is allowed to operate on memory at any given time. That is, when one production finds its triggering condition, it takes control for a moment, disabling all of the other productions. The controlling production manipulates the symbols in memory, and then releases its control, which causes the parallel scan of working memory to recommence.

We have briefly described two characteristics, structure and process, that make production systems examples of physical symbol systems. The third characteristic, control, reveals some additional interesting properties of production systems.

On the one hand, stigmergy is used to control a production system, that is, to choose which production acts at any given time. Stigmergic control occurs when different agents (in this case, productions) do not directly communicate with each other, but conduct indirect communication by modifying a shared environment (Theraulaz & Bonabeau, 1999). Stigmergy has been used to explain how a colony of social insects might coordinate their actions to create a nest (Downing & Jeanne, 1988; Karsai, 1999). The changing structure of the nest elicits different nest-building behaviours; the nest itself controls its own construction. When one insect adds a new piece to the nest, this will change the later behaviour of other insects without any direct communication occurring.

Production system control is stigmergic if the working memory is viewed as being analogous to the insect nest. The current state of the memory causes a
particular production to act. This changes the contents of the memory, which in turn can result in a different production being selected during the next cycle of the architecture.

On the other hand, production system control is usually not completely stigmergic. This is because the stigmergic relationship between working memory and productions is loose enough to produce situations in which conflicts occur. Examples of this type of situation include instances in which more than one production finds its triggering pattern at the same time, or when one production finds its triggering condition present at more than one location in memory at the same time. Such situations must be dealt with by additional control mechanisms. For instance, priorities might be assigned to productions so that in a case where two or more productions were in conflict, only the production with the highest priority would perform its action.

Production systems have provided an architecture—particularly if that architecture is classical in nature—that has been so successful at simulating higher-order cognition that some researchers believe that production systems provide the foundation for a unified theory of cognition (Anderson, 1983; Anderson et al., 2004; Newell, 1990). Production systems illustrate another feature that is also typical of this approach to cognitive science: the so-called classical sandwich (Hurley, 2001).

Imagine a very simple agent that was truly incapable of representation and reasoning. Its interactions with the world would necessarily be governed by a set of reflexes that would convert sensed information directly into action. These reflexes define a sense-act cycle (Pfeifer & Scheier, 1999).

In contrast, a more sophisticated agent could use internal representations to decide upon an action, by reasoning about the consequences of possible actions and choosing the action that was reasoned to be most beneficial (Popper, 1978, p. 354): “While an uncritical animal may be eliminated altogether with its dogmatically held hypotheses, we may formulate our hypotheses, and criticize them. Let our conjectures, our theories die in our stead!” In this second scenario, thinking stands as an intermediary between sensation and action. Such behaviour is not governed by a sense-act cycle, but is instead the product of a sense-think-act cycle (Pfeifer & Scheier, 1999).

Hurley (2001) has argued that the sense-think-act cycle is the stereotypical form of a theory in classical cognitive science; she called this form the classical sandwich. In a typical classical theory, perception can only indirectly inform action, by sending information to be processed by the central representational processes, which in turn decide which action is to be performed.

Production systems exemplify the classical sandwich. The first production systems did not incorporate sensing or acting, in spite of a recognized need to do so. “One problem with psychology’s attempt at cognitive theory has been our persistence
in thinking about cognition without bringing in perceptual and motor processes” (Newell, 1990, p. 15). This was also true of the next generation of production systems, the adaptive control of thought (ACT) architecture (Anderson, 1983). ACT “historically was focused on higher level cognition and not perception or action” (Anderson et al., 2004, p. 1038).

More modern production systems, such as EPIC (executive-process interactive control) (Meyer & Kieras, 1997a, 1997b), have evolved to include sensing and acting. EPIC simulates the performance of multiple tasks and can produce the psychological refractory period (PRP). When two tasks can be performed at the same time, the stimulus onset asynchrony (SOA) between the tasks is the length of time from the start of the first task to the start of the second task. When SOAs are long, the time taken by a subject to make a response is roughly the same for both tasks. However, for SOAs of half a second or less, it takes a longer time to perform the second task than it does to perform the first. This increase in response time for short SOAs is the PRP.

EPIC is an advanced production system. One of its key properties is that productions in EPIC can act in parallel. That is, at any time cycle in EPIC processing, all productions that have matched their conditions in working memory will act to alter working memory. This is important; when multiple tasks are modelled there will be two different sets of productions in action, one for each task. EPIC also includes sensory processors (such as virtual eyes) and motor processors, because actions can constrain task performance. For example, EPIC uses a single motor processor to control two “virtual hands.” This results in interference between two tasks that involve making responses with different hands.

While EPIC (Meyer & Kieras, 1997a, 1997b) explicitly incorporates sensing, acting, and thinking, it does so in a fashion that still exemplifies the classical sandwich. In EPIC, sensing transduces properties of the external world into symbols to be added to working memory. Working memory provides symbolic expressions that guide the actions of motor processors. Thus working memory centralizes the “thinking” that maps sensations onto actions. There are no direct connections between sensing and acting that bypass working memory. EPIC is an example of sense-think-act processing.

Radical embodied cognitive science, which is discussed in Chapter 5, argues that intelligence is the result of situated action; it claims that sense-think-act processing can be replaced by sense-act cycles, and that the rule-governed manipulation of expressions is unnecessary (Chemero, 2009). In contrast, classical researchers claim that production systems that include sensing and acting are sufficient to explain human intelligence and action, and that embodied theories are not necessary (Vera & Simon, 1993).

It follows that there is no need, contrary to what followers of SA [situated action] seem sometimes to claim, for cognitive psychology to adopt a whole new language.
and research agenda, breaking completely from traditional (symbolic) cognitive theories. SA is not a new approach to cognition, much less a new school of cognitive psychology. (Vera & Simon, 1993, p. 46)

We see later in this book that production systems provide an interesting medium that can be used to explore the relationship between classical, connectionist, and embodied cognitive science.

3.11 Weak Equivalence and the Turing Test

There are two fundamentals that follow from accepting the physical symbol system hypothesis (Newell, 1980; Newell & Simon, 1976). First, general human intelligence is the product of rule-governed symbol manipulation. Second, because they are universal machines, any particular physical symbol system can be configured to simulate the behaviour of another physical symbol system.

A consequence of these fundamentals is that digital computers, which are one type of physical symbol system, can simulate another putative member of the same class, human cognition (Newell & Simon, 1961, 1972; Simon, 1969). More than fifty years ago it was predicted “that within ten years most theories in psychology will take the form of computer programs, or of qualitative statements about the characteristics of computer programs” (Simon & Newell, 1958, pp. 7–8). One possible measure of cognitive science’s success is that a leading critic of artificial intelligence has conceded that this particular prediction has been partially fulfilled (Dreyfus, 1992).

There are a number of advantages to using computer simulations to study cognition (Dawson, 2004; Lewandowsky, 1993). The difficulties in converting a theory into a working simulation can identify assumptions that the theory hides. The formal nature of a computer program provides new tools for studying simulated concepts (e.g., proofs of convergence). Programming a theory forces a researcher to provide rigorous definitions of the theory’s components. “Programming is, again like any form of writing, more often than not experimental. One programs, just as one writes, not because one understands, but in order to come to understand.” (Weizenbaum, 1976, p. 108).

However, computer simulation research provides great challenges as well. Chief among these is validating the model, particularly because one universal machine can simulate any other. A common criticism of simulation research is that it is possible to model anything, because modelling is unconstrained:

Just as we may wonder how much the characters in a novel are drawn from real life and how much is artifice, we might ask the same of a model: How much is based on observation and measurement of accessible phenomena, how much is
Based on informed judgment, and how much is convenience? (Oreskes, Shrader-Frechette, & Belitz, 1994, p. 644)

Because of similar concerns, mathematical psychologists have argued that computer simulations are impossible to validate in the same way as mathematical models of behaviour (Estes, 1975; Luce, 1989, 1999). Evolutionary biologist John Maynard Smith called simulation research “fact free science” (Mackenzie, 2002).

Computer simulation researchers are generally puzzled by such criticisms, because their simulations of cognitive phenomena must conform to a variety of challenging constraints (Newell, 1980, 1990; Pylyshyn, 1984). For instance, Newell’s (1980, 1990) production system models aim to meet a number of constraints that range from behavioural (flexible responses to environment, goal-oriented, operate in real time) to biological (realizable as a neural system, develop via embryological growth processes, arise through evolution).

In validating a computer simulation, classical cognitive science becomes an intrinsically comparative discipline. Model validation requires that theoretical analyses and empirical observations are used to evaluate both the relationship between a simulation and the subject being simulated. In adopting the physical symbol system hypothesis, classical cognitive scientists are further committed to the assumption that this relation is complex, because it can be established (as argued in Chapter 2) at many different levels (Dawson, 1998; Marr, 1982; Pylyshyn, 1984). Pylyshyn has argued that model validation can take advantage of this and proceed by imposing severe empirical constraints. These empirical constraints involve establishing that a model provides an appropriate account of its subject at the computational, algorithmic, and architectural levels of analysis. Let us examine this position in more detail.

First, consider a relationship between model and subject that is not listed above—a relationship at the implementational level of analysis. Classical cognitive science’s use of computer simulation methodology is a tacit assumption that the physical structure of its models does not need to match the physical structure of the subject being modelled.

The basis for this assumption is the multiple realization argument that we have already encountered. Cognitive scientists describe basic information processes in terms of their functional nature and ignore their underlying physicality. This is because the same function can be realized in radically different physical media. For instance, AND-gates can be created using hydraulic channels, electronic components, or neural circuits (Hillis, 1998). If hardware or technology were relevant—if the multiple realization argument was false—then computer simulations of cognition would be absurd. Classical cognitive science ignores the physical when models are validated. Let us now turn to the relationships between models and subjects that classical cognitive science cannot and does not ignore.
In the most abstract sense, both a model and a modelled agent can be viewed as opaque devices, black boxes whose inner workings are invisible. From this perspective, both are machines that convert inputs or stimuli into outputs or responses; their behaviour computes an input-output function (Ashby, 1956, 1960). Thus the most basic point of contact between a model and its subject is that the input-output mappings produced by one must be identical to those produced by the other. Establishing this fact is establishing a relationship between model and subject at the computational level.

To say that a model and subject are computing the same input-output function is to say that they are weakly equivalent. It is a weak equivalence because it is established by ignoring the internal workings of both model and subject. There are an infinite number of different algorithms for computing the same input-output function (Johnson-Laird, 1983). This means that weak equivalence can be established between two different systems that use completely different algorithms. Weak equivalence is not concerned with the possibility that two systems can produce the right behaviours but do so for the wrong reasons.

Weak equivalence is also sometimes known as Turing equivalence. This is because weak equivalence is at the heart of a criterion proposed by computer pioneer Alan Turing, to determine whether a computer program had achieved intelligence (Turing, 1950). This criterion is called the Turing test.

Turing (1950) believed that a device's ability to participate in a meaningful conversation was the strongest test of its general intelligence. His test involved a human judge conducting, via teletype, a conversation with an agent. In one instance, the agent was another human. In another, the agent was a computer program. Turing argued that if the judge could not correctly determine which agent was human then the computer program must be deemed to be intelligent. A similar logic was subscribed to by Descartes (2006). Turing and Descartes both believed in the power of language to reveal intelligence; however, Turing believed that machines could attain linguistic power, while Descartes did not.

A famous example of the application of the Turing test is provided by a model of paranoid schizophrenia, PARRY (Kosslyn, Ball, & Reiser, 1978). This program interacted with a user by carrying on a conversation—it was a natural language communication program much like the earlier ELIZA program (Weizenbaum, 1966). However, in addition to processing the structure of input sentences, PARRY also computed variables related to paranoia: fear, anger, and mistrust. PARRY's responses were thus affected not only by the user's input, by also by its evolving affective states. PARRY's contributions to a conversation became more paranoid as the interaction was extended over time.

A version of the Turing test was used to evaluate PARRY's performance (Colby et al., 1972). Psychiatrists used teletypes to interview PARRY as well as human
paranoids. Forty practising psychiatrists read transcripts of these interviews in order to distinguish the human paranoids from the simulated ones. They were only able to do this at chance levels. PARRY had passed the Turing test: “We can conclude that psychiatrists using teletyped data do not distinguish real patients from our simulation of a paranoid patient” (p. 220).

The problem with the Turing test, though, is that in some respects it is too easy to pass. This was one of the points of the pioneering conversation-making program, ELIZA (Weizenbaum, 1966), which was developed to engage in natural language conversations. Its most famous version, DOCTOR, modelled the conversational style of an interview with a humanistic psychotherapist. ELIZA's conversations were extremely compelling. “ELIZA created the most remarkable illusion of having understood the minds of the many people who conversed with it” (Weizenbaum, 1976, p. 189). Weizenbaum was intrigued by the fact that “some subjects have been very hard to convince that ELIZA is not human. This is a striking form of Turing's test” (Weizenbaum, 1966, p. 42).

However, ELIZA's conversations were not the product of natural language understanding. It merely parsed incoming sentences, and then put fragments of these sentences into templates that were output as responses. Templates were ranked on the basis of keywords that ELIZA was programmed to seek during a conversation; this permitted ELIZA to generate responses rated as being highly appropriate. “A large part of whatever elegance may be credited to ELIZA lies in the fact that ELIZA maintains the illusion of understanding with so little machinery” (Weizenbaum, 1966, p. 43).

Indeed, much of the apparent intelligence of ELIZA is a contribution of the human participant in the conversation, who assumes that ELIZA understands its inputs and that even strange comments made by ELIZA are made for an intelligent reason.

The ‘sense’ and the continuity the person conversing with ELIZA perceives is supplied largely by the person himself. He assigns meanings and interpretations to what ELIZA ‘says’ that confirm his initial hypothesis that the system does understand, just as he might do with what a fortune-teller says to him.

(Weizenbaum, 1976, p. 190)

Weizenbaum believed that natural language understanding was beyond the capability of computers, and also believed that ELIZA illustrated this belief. However, ELIZA was received in a fashion that Weizenbaum did not anticipate, and which was opposite to his intent. He was so dismayed that he wrote a book that served as a scathing critique of artificial intelligence research (Weizenbaum, 1976, p. 2): “My own shock was administered not by any important political figure in establishing his philosophy of science, but by some people who insisted on misinterpreting a piece of work I had done.”
The ease with which ELIZA was misinterpreted—that is, the ease with which it passed a striking form of Turing’s test—caused Weizenbaum (1976) to question most research on the computer simulation of intelligence. Much of Weizenbaum’s concern was rooted in AI’s adoption of Turing’s (1950) test as a measure of intelligence.

An entirely too simplistic notion of intelligence has dominated both popular and scientific thought, and this notion is, in part, responsible for permitting artificial intelligence’s perverse grand fantasy to grow. (Weizenbaum, 1976, p. 203)

However, perhaps a more reasoned response would be to adopt a stricter means of evaluating cognitive simulations. While the Turing test has had more than fifty years of extreme influence, researchers are aware of its limitations and have proposed a number of ways to make it more sensitive (French, 2000).

For instance, the Total Turing Test (French, 2000) removes the teletype and requires that a simulation of cognition be not only conversationally indistinguishable from a human, but also physically indistinguishable. Only a humanoid robot could pass such a test, and only do so by not only speaking but also behaving (in very great detail) in ways indistinguishable from a human. A fictional version of the Total Turing Test is the Voight-Kampff scale described in Dick’s (1968) novel Do Androids Dream of Electric Sheep? This scale used behavioural measures of empathy, including pupil dilation, to distinguish humans from androids.

3.12 Towards Strong Equivalence

The Turing test has had a long, influential history (French, 2000). However, many would agree that it is flawed, perhaps because it is too easily passed. As a consequence, some have argued that artificial intelligence research is very limited (Weizenbaum, 1976). Others have argued for more stringent versions of the Turing test, such as the Total Turing Test.

Classical cognitive science recognizes that the Turing test provides a necessary, but not a sufficient, measure of a model’s validity. This is because it really only establishes weak equivalence, by collecting evidence that two systems are computationally equivalent. It accomplishes this by only examining the two devices at the level of the input-output relationship. This can only establish weak equivalence, because systems that use very different algorithms and architectures can still compute the same function.

Classical cognitive science has the goal of going beyond weak equivalence. It attempts to do so by establishing additional relationships between models and subjects, identities between both algorithms and architectures. This is an attempt to establish what is known as strong equivalence (Pylyshyn, 1984). Two systems are said to be strongly equivalent if they compute the same input-output function (i.e.,
if they are weakly equivalent), accomplish this with the same algorithm, and bring this algorithm to life with the same architecture. Cognitive scientists are in the business of making observations that establish the strong equivalence of their models to human thinkers.

Classical cognitive science collects these observations by measuring particular behaviours that are unintended consequences of information processing, and which can therefore reveal the nature of the algorithm that is being employed. Newell and Simon (1972) named these behaviours second-order effects; in Chapter 2 these behaviours were called artifacts, to distinguish them from the primary or intended responses of an information processor. In Chapter 2, I discussed three general classes of evidence related to artifactual behaviour: intermediate state evidence, relative complexity evidence, and error evidence.

Note that although similar in spirit, the use of these three different types of evidence to determine the relationship between the algorithms used by model and subject is not the same as something like the Total Turing Test. Classical cognitive science does not require physical correspondence between model and subject. However, algorithmic correspondences established by examining behavioural artifacts put much stronger constraints on theory validation than simply looking for stimulus-response correspondences. To illustrate this, let us consider some examples of how intermediate state evidence, relative complexity evidence, and error evidence can be used to validate models.

One important source of information that can be used to validate a model is intermediate state evidence (Pylyshyn, 1984). Intermediate state evidence involves determining the intermediate steps that a symbol manipulator takes to solve a problem, and then collecting evidence to determine whether a modelled subject goes through the same intermediate steps. Intermediate state evidence is notoriously difficult to collect, because human information processors are black boxes—we cannot directly observe internal cognitive processing. However, clever experimental paradigms can be developed to permit intermediate states to be inferred.

A famous example of evaluating a model using intermediate state evidence is found in some classic and pioneering research on human problem solving (Newell & Simon, 1972). Newell and Simon collected data from human subjects as they solved problems; their method of data collection is known as protocol analysis (Ericsson & Simon, 1984). In protocol analysis, subjects are trained to think out loud as they work. A recording of what is said by the subject becomes the primary data of interest.

The logic of collecting verbal protocols is that the thought processes involved in active problem solving are likely to be stored in a person's short-term memory (STM), or working memory. Cognitive psychologists have established that items stored in such a memory are stored as an articulatory code that permits verbalization to
maintain the items in memory (Baddeley, 1986, 1990; Conrad, 1964a, 1964b; Waugh & Norman, 1965). As a result, asking subjects to verbalize their thinking steps is presumed to provide accurate access to current cognitive processing, and to do so with minimal disruption. “Verbalization will not interfere with ongoing processes if the information stored in STM is encoded orally, so that an articulatory code can readily be activated”” (Ericsson & Simon, 1984, p. 68).

In order to study problem solving, Newell and Simon (1972) collected verbal protocols for problems that were difficult enough to engage subjects and generate interesting behaviour, but simple enough to be solved. For instance, when a subject was asked to decode the cryptarithmetic problem DONALD + GERALD = ROBERT after being told that D = 5, they solved the problem in twenty minutes and produced a protocol that was 2,186 words in length.

The next step in the study was to create a problem behaviour graph from a subject's protocol. A problem behaviour graph is a network of linked nodes. Each node represents a state of knowledge. For instance, in the cryptarithmetic problem such a state might be the observation that “R is odd.” A horizontal link from a node to a node on its right represents the application of an operation that changed the state of knowledge. An example operation might be “Find a column that contains a letter of interest and process that column.” A vertical link from a node to a node below represents backtracking. In many instances, a subject would reach a dead end in a line of thought and return to a previous state of knowledge in order to explore a different approach. The 2,186-word protocol produced a problem behaviour graph that consisted of 238 different nodes.

The initial node in a problem behaviour graph represents a subject's starting state of knowledge when given a problem. A node near the end of the problem behaviour graph represents the state of knowledge when a solution has been achieved. All of the other nodes represent intermediate states of knowledge. Furthermore, in Newell and Simon's (1972) research, these intermediate states represent very detailed elements of knowledge about the problem as it is being solved.

The goal of the simulation component of Newell and Simon's (1972) research was to create a computer model that would generate its own problem behaviour graph. The model was intended to produce a very detailed mimicry of the subject's behaviour—it was validated by examining the degree to which the simulation's problem behaviour graph matched the graph created for the subject. The meticulous nature of such intermediate state evidence provided additional confidence for the use of verbal protocols as scientific data. “For the more information conveyed in their responses, the more difficult it becomes to construct a model that will produce precisely those responses adventitiously—hence the more confidence we can place in a model that does predict them” (Ericsson & Simon, 1984, p. 7).
Newell and Simon (1972) created a computer simulation by examining a subject's problem behaviour graph, identifying the basic processes that it revealed in its links between nodes, and coding each of these processes as a production system. Their model developed from the protocol for the DONALD + GERALD = ROBERT problem consisted of only 14 productions. The behaviour of this fairly small program was able to account for 75 to 80 percent of the human subject's problem behaviour graph. "All of this analysis shows how a verbal thinking-aloud protocol can be used as the raw material for generating and testing a theory of problem solving behavior" (Newell & Simon, 1972, p. 227).

The contribution of Newell and Simon's (1972) research to classical cognitive science is impossible to overstate. One of their central contributions was to demonstrate that human problem solving could be characterized as searching through a problem space. A problem space consists of a set of knowledge states—starting state, one or more goal states, and a potentially large number of intermediate states—that each represent current knowledge about a problem. A link between two knowledge states shows how the application of a single rule can transform the first state into the second. A problem behaviour graph is an example of a problem space. Searching the problem space involves finding a route—a sequence of operations—that will transform the initial state into a goal state. From this perspective, problem solving becomes the domain of control: finding as efficiently as possible an acceptable sequence of problem-solving operations. An enormous number of different search strategies exist (Knuth, 1997; Nilsson, 1980); establishing the strong equivalence of a problem-solving model requires collecting evidence (e.g., using protocol analysis) to ensure that the same search or control strategy is used by both model and agent.

A second kind of evidence that is used to investigate the validity of a model is relative complexity evidence (Pylyshyn, 1984). Relative complexity evidence generally involves examining the relative difficulty of problems, to see whether the problems that are hard (or easy) for a model are the same problems that are hard (or easy) for a modelled subject. The most common kind of relative complexity evidence collected by cognitive scientists is response latency (Luce, 1986; Posner, 1978). It is assumed that the time taken for a system to generate a response is an artificial behaviour that can reveal properties of an underlying algorithm and be used to examine the algorithmic relationship between model and subject.

One domain in which measures of response latency have played an important role is the study of visual cognition (Kosslyn & Osherson, 1995; Pinker, 1985). Visual cognition involves solving information processing problems that involve spatial relationships or the spatial layout of information. It is a rich domain of study because it seems to involve qualitatively different kinds of information processing: the data-driven or preattentive detection of visual features (Marr, 1976;
Richards, 1988; Treisman, 1985), top-down or high-level cognition to link combinations of visual features to semantic interpretations or labels (Jackendoff, 1983, 1987; Treisman, 1986, 1988), and processing involving visual attention or visual routines that include both data-driven and top-down characteristics, and which serve as an intermediary between feature detection and object recognition (Cooper & Shepard, 1973a, 1973b; Ullman, 1984; Wright, 1998).

Visual search tasks are frequently used to study visual cognition. In such a task, a subject is usually presented with a visual display consisting of a number of objects. In the odd-man-out version of this task, in one half of the trials one of the objects (the target) is different from all of the other objects (the distracters). In the other half of the trials, the only objects present are distracters. Subjects have to decide as quickly and accurately as possible whether a target is present in each display. The dependent measures in such tasks are search latency functions, which represent the time required to detect the presence or absence of a target as a function of the total number of display elements.

Pioneering work on visual search discovered the so-called pop-out effect: the time required to detect the presence of a target that is characterized by one of a small number of unique features (e.g., colour, orientation, contrast, motion) is largely independent of the number of distractor elements in a display, producing a search latency function that is essentially flat (Treisman & Gelade, 1980). This is because, regardless of the number of elements in the display, when the target is present it seems to pop out of the display, bringing itself immediately to attention. Notice how the target pops out of the display illustrated in Figure 3-11.

Figure 3-11. Unique features pop out of displays, regardless of display size.

In contrast, the time to detect a target defined by a unique combination of features generally increases with the number of distractor items, producing search latency functions with positive slopes. Figure 3-12 illustrates visual search in objects that are either connected or unconnected (Dawson & Thibodeau, 1998); connectedness
is a property that is not local, but is only defined by relations between multiple features (Minsky & Papert, 1988). The larger the number of display items, the longer it takes to find the target when it is present in the display. Is there a target in Figure 3-12? If so, is it harder to find than the one that was present in Figure 3-11?

Search latency results as those described above, which revealed that some objects pop out but others do not, formed the basis for feature integration theory (Treisman, 1985, 1986, 1988; Treisman & Gelade, 1980; Treisman & Gormican, 1988; Treisman, Sykes, & Gelade, 1977). Feature integration theory is a multistage account of visual cognition. In the first state, preattentive processors register the locations of a small set of primitive visual features on independent feature maps. These maps represent a small number of properties (e.g., orientation, colour, contrast movement) that also appear to be transduced by early neural visual detectors (Livingstone & Hubel, 1988). If such a feature is unique to a display, then it will be the only active location in its feature map. This permits pop out to occur, because the location of the unique, primitive feature is preattentively available.

Unique combinations of features do not produce unique activity in a single feature map and therefore cannot pop out. Instead, they require additional processing in order to be detected. First, attentional resources must be used to bring the various independent feature maps into register with respect to a master map of locations. This master map of locations will indicate what combinations of features coexist at each location in the map. Second, a “spotlight” of attention is used to scan the master map of locations in search of a unique object. Because this attentional spotlight can only process a portion of the master map at any given time, and because it must be scanned from location to location on the master map, it takes longer for unique combinations of features to be found. Furthermore, the search of the master map will become longer and longer as more of its locations are filled,
explaining why the latency to detect unique feature combinations is affected by the number of distractors present.

Relative complexity evidence can also be used to explore some of the components of feature integration theory. For example, several researchers have proposed models of how the attentional spotlight is shifted to detect targets in a visual search task (Fukushima, 1986; Gerrissen, 1991; Grossberg, 1980; Koch & Ullman, 1985; LaBerge, Carter, & Brown, 1992; Sandon, 1992). While the specific details of these models differ, their general structure is quite similar. First, these models represent the display being searched as an array of processors whose activities encode the visual distinctiveness of the location that each processor represents (i.e., how different it is in appearance relative to its neighbours). Second, these processors engage in a winner-take-all (WTA) competition (Feldman & Ballard, 1982) to identify the most distinctive location. This competition is defined by lateral inhibition: each processor uses its activity as an inhibitory signal in an attempt to reduce the activity of its neighbours. Third, the display element at the winning location is examined to see whether or not it is the target. If it is, the search stops. If it is not, activity at this location either decays or is inhibited (Klein, 1988), and a new WTA competition is used to find the next most distinctive location in the display.

This type of model provides a straightforward account of search latency functions obtained for targets defined by unique conjunctions of features. They also lead to a unique prediction: if inhibitory processes are responsible for directing the shift of the attentional spotlight, then search latency functions should be affected by the overall adapting luminance of the display. This is because there is a greater degree of inhibition during the processing of bright visual displays than there is for dimmer displays (Barlow, Fitzhugh, & Kuffler, 1957; Derrington & Lennie, 1982; Ransom-Hogg & Spillmann, 1980; Rohaly & Buchsbaum, 1989).

A visual search study was conducted to test this prediction (Dawson & Thibodeau, 1998). Modifying a paradigm used to study the effect of adaptive luminance on motion perception (Dawson & Di Lollo, 1990), Dawson and Thibodeau (1998) had subjects perform a visual search task while viewing the displays through neutral density filters that modified display luminance while not affecting the relative contrast of elements. There were two major findings that supported the kinds of models of attentional shift described above. First, when targets pop out, the response latency of subjects was not affected by adaptive luminance. This is consistent with feature integration theory, in the sense that a shifting attentional spotlight is not required for pop out to occur. Second, for targets that did not pop out, search latency functions were affected by the level of adaptive luminance. For darker displays, both the intercept and the slope of the search latency functions increased significantly. This is consistent with the hypothesis that this manipulation interferes with the inhibitory processes that guide shifts of attention.
A third approach to validating a model involves the use of error evidence. This approach assumes that errors are artifacts, in the sense that they are a natural consequence of an agent’s information processing, and that they are not a deliberate or intended product of this processing.

One source of artifactual errors is the way information processing can be constrained by limits on internal resources (memory or attention) or by external demands (the need for real time responses). These restrictions on processing produce bounded rationality (Simon, 1982). Another reason for artifactual errors lies in the restrictions imposed by the particular structure-process pairing employed by an information processor. “A tool too gains its power from the fact that it permits certain actions and not others. For example, a hammer has to be rigid. It can therefore not be used as a rope” (Weizenbaum, 1976, p. 37). Like a tool, a particular structure-process pairing may not be suited for some tasks and therefore produces errors when faced with them.

One example of the importance of error evidence is found in the large literature on human, animal, and robot navigation (Cheng, 2005; Cheng & Newcombe, 2005; Healy, 1998; Jonsson, 2002; Milford, 2008). How do organisms find their place in the world? One approach to answering this question is to set up small, manageable indoor environments. These “arenas” can provide a variety of cues to animals that learn to navigate within them. If an agent is reinforced for visiting a particular location, what cues does it use to return to this place?

One paradigm for addressing this question is the reorientation task invented by Ken Cheng (1986). In the reorientation task, an agent is typically placed within a rectangular arena. Reinforcement is typically provided at one of the corner locations in the arena. That is, the agent is free to explore the arena, and eventually finds a reward at a location of interest—it learns that this is the “goal location.” The agent is then removed from the arena, disoriented, and returned to an (often different) arena, with the task of using the available cues to relocate the goal. Of particular interest are experimental conditions in which the arena has been altered from the one in which the agent was originally trained.

An arena that is used in the reorientation task can provide two different kinds of navigational information: geometric cues and feature cues (Cheng & Newcombe, 2005). Geometric cues are relational, while feature cues are not.

A geometric property of a surface, line, or point is a property it possesses by virtue of its position relative to other surfaces, lines, and points within the same space. A non-geometric property is any property that cannot be described by relative position alone. (Gallistel, 1990, p. 212)

In a rectangular arena, metric properties (e.g., wall lengths, angles between walls) combined with an agent’s distinction between left and right (e.g., the long wall is to the left of the short wall) provide geometric cues. Non-geometric cues or feature
cues can be added as well. For instance, one arena wall can have a different colour than the others (Cheng, 1986), or different coloured patterns can be placed at each corner of the arena (Kelly, Spetch, & Heth, 1998).

One question of interest concerns the relative contributions of these different cues for reorientation. This is studied by seeing how the agent reorients after it has been returned to an arena in which cues have been altered. For example, the feature cues might have been moved to new locations. This places feature cues in conflict with geometric cues. Will the agent move to a location defined by geometric information, or will it move to a different location indicated by feature information? Extensive use of the reorientation task has revealed some striking regularities.

Some of the most interesting regularities found in the reorientation task pertain to a particular error in reorientation. In an arena with no unique feature cues (no unique wall colour, no unique pattern at each corner), geometric cues are the only information available for reorienting. However, geometric cues cannot uniquely specify a goal location in a rectangular arena. This is because the geometric cues at the goal location (e.g., 90° angle, shorter wall to the left and longer wall to the right) are identical to the geometric cues present at the diagonally opposite corner (often called the rotational location). Under these conditions, the agent will produce rotational error (Cheng, 1986, 2005). When rotational error occurs, the trained agent goes to the goal location at above-chance levels; however, the animal goes to the rotational location equally often. Rotational error is usually taken as evidence that the agent is relying upon the geometric properties of the environment.

When feature cues are present in a rectangular arena, a goal location can be uniquely specified. In fact, when cues are present, an agent should not even need to pay attention to geometric cues, because these cues are not relevant. However, evidence suggests that geometric cues still influence behaviour even when such cues are not required to solve the task.

First, in some cases subjects continue to make some rotational errors even when feature cues specify the goal location (Cheng, 1986; Hermer & Spelke, 1994). Second, when feature cues present during training are removed from the arena in which reorientation occurs, subjects typically revert to generating rotational error (Kelly, Spetch, and Heth, 1998; Sovrano, Bisazza, & Vallortigara, 2003). Third, in studies in which local features are moved to new locations in the new arena, there is a conflict between geometric and feature cues. In this case, reorientation appears to be affected by both types of cues. The animals will not only increase their tendency to visit the corner marked by the feature cues that previously signaled the goal, but also produce rotational error for two other locations in the arena (Brown, Spetch, & Hurd, 2007; Kelly, Spetch, and Heth, 1998).

Rotational error is an important phenomenon in the reorientation literature, and it is affected by a complex interaction between geometric and feature cues. A
growing variety of models of reorientation are appearing in the literature, including models consistent with the symbol-manipulating fundamental of classical cognitive science (Cheng, 1986; Gallistel, 1990), neural network models that are part of connectionist cognitive science (Dawson et al., 2010), and behaviour-based robots that are the domain of embodied cognitive science (Dawson, Dupuis, & Wilson, 2010; Nolfi, 2002). All of these models have two things in common. First, they can produce rotational error and many of its nuances. Second, this error is produced as a natural byproduct of a reorientation algorithm; the errors produced by the models are used in aid of their validation.

3.13 The Impenetrable Architecture

Classical cognitive scientists often develop theories in the form of working computer simulations. These models are validated by collecting evidence that shows they are strongly equivalent to the subjects or phenomena being modelled. This begins by first demonstrating weak equivalence, that both model and subject are computing the same input-output function. The quest for strong equivalence is furthered by using intermediate state evidence, relative complexity evidence, and error evidence to demonstrate, in striking detail, that both model and subject are employing the same algorithm.

However, strong equivalence can only be established by demonstrating an additional relationship between model and subject. Not only must model and subject be employing the same algorithm, but both must also be employing the same primitive processes. Strong equivalence requires architectural equivalence.

The primitives of a computer simulation are readily identifiable. A computer simulation should be a collection of primitives that are designed to generate a behaviour of interest (Dawson, 2004). In order to create a model of cognition, one must define the basic properties of a symbolic structure, the nature of the processes that can manipulate these expressions, and the control system that chooses when to apply a particular rule, operation, or process. A model makes these primitive characteristics explicit. When the model is run, its behaviour shows what these primitives can produce.

While identifying a model’s primitives should be straightforward, determining the architecture employed by a modelled subject is far from easy. To illustrate this, let us consider research on mental imagery.

Mental imagery is a cognitive phenomenon in which we experience or imagine mental pictures. Mental imagery is often involved in solving spatial problems (Kosslyn, 1980). For instance, imagine being asked how many windows there are on the front wall of the building in which you live. A common approach to answering this question would be to imagine the image of this wall and to inspect the image,
mentally counting the number of windows that are displayed in it. Mental imagery is also crucially important for human memory (Paivio, 1969, 1971, 1986; Yates, 1966): we are better at remembering items if we can create a mental image of them. Indeed, the construction of bizarre mental images, or of images that link two or more items together, is a standard tool of the mnemonic trade (Lorayne, 1985, 1998, 2007; Lorayne & Lucas, 1974).

An early achievement of the cognitive revolution in psychology (Miller, 2003; Vauclair & Perret, 2003) was a rekindled interest in studying mental imagery, an area that had been neglected during the reign of behaviourism (Paivio, 1971, 1986). In the early stages of renewed imagery research, traditional paradigms were modified to solidly establish that concept imageability was a key predictor of verbal behaviour and associative learning (Paivio, 1969). In later stages, new paradigms were invented to permit researchers to investigate the underlying nature of mental images (Kosslyn, 1980; Shepard & Cooper, 1982).

For example, consider the relative complexity evidence obtained using the mental rotation task (Cooper & Shepard, 1973a, 1973b; Shepard & Metzler, 1971). In this task, subjects are presented with a pair of images. In some instances, the two images are of the same object. In other instances, the two images are different (e.g., one is a mirror image of the other). The orientation of the images can also be varied—for instance, they can be rotated to different degrees in the plane of view. The angular disparity between the two images is the key independent variable. A subject’s task is to judge whether the images are the same or not; the key dependent measure is the amount of time required to respond.

In order to perform the mental rotation task, subjects first construct a mental image of one of the objects, and then imagine rotating it to the correct orientation to enable them to judge whether it is the same as the other object. The standard finding in this task is that there is a linear relationship between response latency and the amount of mental rotation that is required. From these results it has been concluded that “the process of mental rotation is an analog one in that intermediate states in the process have a one-to-one correspondence with intermediate stages in the external rotation of an object” (Shepard & Cooper, 1982, p. 185). That is, mental processes rotate mental images in a holistic fashion, through intermediate orientations, just as physical processes can rotate real objects.

Another source of relative complexity evidence concerning mental imagery is the image scanning task (Kosslyn, 1980; Kosslyn, Ball, & Reisler, 1978). In the most famous version of this task, subjects are first trained to create an accurate mental image of an island map on which seven different locations are marked. Then subjects are asked to construct this mental image, focusing their attention at one of the locations. They are then provided with a name, which may or may not be one of the other map locations. If the name is of another map location, then subjects...
are instructed to scan across the image to it, pressing a button when they arrive at the second location.

In the map-scanning version of the image-scanning task, the dependent variable was the amount of time from the naming of the second location to a subject's button press, and the independent variable was the distance on the map between the first and second locations. The key finding was that there was nearly a perfectly linear relationship between latency and distance (Kosslyn Ball, & Reisler, 1978): an increased distance led to an increased response latency, suggesting that the image had spatial extent, and that it was scanned at a constant rate.

The scanning experiments support the claim that portions of images depict corresponding portions of the represented objects, and that the spatial relations between portions of the image index the spatial relations between the corresponding portions of the imaged objects. (Kosslyn, 1980, p. 51)

The relative complexity evidence obtained from tasks like mental rotation and image scanning provided the basis for a prominent account of mental imagery known as the depictive theory (Kosslyn, 1980, 1994; Kosslyn, Thompson, & Ganis, 2006). This theory is based on the claim that mental images are not merely internal representations that describe visuospatial information (as would be the case with words or with logical propositions), but instead depict this information because the format of an image is quasi-pictorial. That is, while a mental image is not claimed to literally be a picture in the head, it nevertheless represents content by resemblance.

There is a correspondence between parts and spatial relations of the representation and those of the object; this structural mapping, which confers a type of resemblance, underlies the way images convey specific content. In this respect images are like pictures. Unlike words and symbols, depictions are not arbitrarily paired with what they represent. (Kosslyn, Thompson, & Ganis, 2006, p. 44)

The depictive theory specifies primitive properties of mental images, which have sometimes been called privileged properties (Kosslyn, 1980). What are these primitives? One is that images occur in a spatial medium that is functionally equivalent to a coordinate space. A second is that images are patterns that are produced by activating local regions of this space to produce an “abstract spatial isomorphism” (Kosslyn, 1980, p. 33) between the image and what it represents. This isomorphism is a correspondence between an image and a represented object in terms of their parts as well as spatial relations amongst these parts. A third is that images not only depict spatial extent, they also depict properties of visible surfaces such as colour and texture.

These privileged properties are characteristic of the format mental images—the structure of images as symbolic expressions. When such a structure is paired with particular primitive processes, certain types of questions are easily answered. These
processes are visual in nature: for instance, mental images can be scanned, inspected at different apparent sizes, or rotated. The coupling of such processes with the depictive structure of images is well-suited to solving visuospatial problems. Other structure-process pairings—in particular, logical operations on propositional expressions that describe spatial properties (Pylyshyn, 1973)—do not make spatial information explicit and arguably will not be as adept at solving visuospatial problems. Kosslyn (1980, p. 35) called the structural properties of images privileged because their possession “[distinguishes] an image from other forms of representation.”

That the depictive theory makes claims about the primitive properties of mental images indicates quite clearly that it is an account of cognitive architecture. That it is a theory about architecture is further supported by the fact that the latest phase of imagery research has involved the supplementing behavioural data with evidence concerning the cognitive neuroscience of imagery (Kosslyn, 1994; Kosslyn et al., 1995; Kosslyn et al., 1999; Kosslyn, Thompson, & Alpert, 1997; Kosslyn, Thompson, & Ganis, 2006). This research has attempted to ground the architectural properties of images into topographically organized regions of the cortex.

Computer simulation has proven to be a key medium for evaluating the depictive theory of mental imagery. Beginning with work in the late 1970s (Kosslyn & Shwartz, 1977), the privileged properties of mental images have been converted into a working computer model (Kosslyn, 1980, 1987, 1994; Kosslyn et al., 1984; Kosslyn et al., 1985). In general terms, over time these models represent an elaboration of a general theoretical structure: long-term memory uses propositional structures to store spatial information. Image construction processes convert this propositional information into depictive representations on a spatial medium that enforces the primitive structural properties of images. Separate from this medium are primitive processes that operate on the depicted information (e.g., scan, inspect, interpret). This form of model has shown that the privileged properties of images that define the depictive theory are sufficient for simulating a wide variety of the regularities that govern mental imagery.

The last few paragraphs have introduced Kosslyn’s (e.g., 1980) depictive theory, its proposals about the privileged properties of mental images, and the success that computer simulations derived from this theory have had at modelling behavioural results. All of these topics concern statements about primitives in the domain of a theory or model about mental imagery. Let us now turn to one issue that has not yet been addressed: the nature of the primitives employed by the modelled subject, the human imager.

The status of privileged properties espoused by the depictive theory has been the subject of a decades-long imagery debate (Block, 1981; Tye, 1991). At the heart of the imagery debate is a basic question: are the privileged properties parts of the architecture or not? The imagery debate began with the publication of a seminal
paper (Pylyshyn, 1973), which proposed that the primitive properties of images were not depictive, but were instead descriptive properties based on a logical or propositional representation. This position represents the basic claim of the propositional theory, which stands as a critical alternative to the depictive theory. The imagery debate continues to the present day; propositional theory’s criticism of the depictive position has been prolific and influential (Pylyshyn, 1981a, 1981b, 1984, 2003a, 2003b, 2003c, 2007).

The imagery debate has been contentious, has involved a number of different subtle theoretical arguments about the relationship between theory and data, and has shown no signs of being clearly resolved. Indeed, some have argued that it is a debate that is cannot be resolved, because it is impossible to identify data that is appropriate to differentiate the depictive and propositional theories (Anderson, 1978). In this section, the overall status of the imagery debate is not of concern. We are instead interested in a particular type of evidence that has played an important role in the debate: evidence concerning cognitive penetrability (Pylyshyn, 1980, 1984, 1999).

Recall from the earlier discussion of algorithms and architecture that Newell (1990) proposed that the rate of change of various parts of a physical symbol system would differ radically depending upon which component was being examined. Newell observed that data should change rapidly, stored programs should be more enduring, and the architecture that interprets stored programs should be even more stable. This is because the architecture is wired in. It may change slowly (e.g., in human cognition because of biological development), but it should be the most stable information processing component. When someone claims that they have changed their mind, we interpret this as meaning that they have updated their facts, or that they have used a new approach or strategy to arrive at a conclusion. We don’t interpret this as a claim that they have altered their basic mental machinery—when we change our mind, we don’t change our cognitive architecture!

The cognitive penetrability criterion (Pylyshyn, 1980, 1984, 1999) is an experimental paradigm that takes advantage of the persistent “wired in” nature of the architecture. If some function is part of the architecture, then it should not be affected by changes in cognitive content—changing beliefs should not result in a changing architecture. The architecture is cognitively impenetrable. In contrast, if some function changes because of a change in content that is semantically related to the function, then this is evidence that it is not part of the architecture.

If a system is cognitively penetrable then the function it computes is sensitive, in a semantically coherent way, to the organism’s goals and beliefs, that is, it can be altered in a way that bears some logical relation to what the person knows.

(Pylyshyn, 1999, p. 343)

The architecture is not cognitively penetrable.
Cognitive penetrability provides a paradigm for testing whether a function of interest is part of the architecture or not. First, some function is measured as part of a pre-test. For example, consider Figure 3-13, which presents the Müller-Lyer illusion, which was discovered in 1889 (Gregory, 1978). In a pre-test, it would be determined whether you experience this illusion. Some measurement would be made to determine whether you judge the horizontal line segment of the top arrow to be longer than the horizontal line segment of the bottom arrow.

Second, a strong manipulation of a belief related to the function that produces the Müller-Lyer illusion would be performed. You, as a subject, might be told that the two horizontal line segments were equal in length. You might be given a ruler, and asked to measure the two line segments, in order to convince yourself that your experience was incorrect and that the two lines were of the same length.

Third, a post-test would determine whether you still experienced the illusion. Do the line segments still appear to be of different length, even though you are armed with the knowledge that this appearance is false? This illusion has had such a long history because its appearance is not affected by such cognitive content. The mechanism that is responsible for the Müller-Lyer illusion is cognitively impenetrable.

This paradigm has been applied to some of the standard mental imagery tasks in order to show that some of the privileged properties of images are cognitively penetrable and therefore cannot be part of the architecture. For instance, in his 1981 dissertation, Liam Bannon examined the map scanning task for cognitive penetrability (for methodological details, see Pylyshyn, 1981a). Bannon reasoned that the instructions given to subjects in the standard map scanning study (Kosslyn, Ball, & Reiser, 1978) instilled a belief that image scanning was like scanning a picture. Bannon was able to replicate the Kosslyn, Ball, & Reiser results in one condition. However, in other conditions the instructions were changed so that the images had to be scanned to answer a question, but no beliefs about scanning were instilled. In one study, Bannon had subjects shift attention from the first map location to the second (named) location, and then judge the compass direction from the second
location to the first. In this condition, the linearly increasing relationship between
distance and time disappeared. Image scanning appears to be cognitively penetrable,
challenging some of the architectural claims of depictive theory. “Images can
be examined without the putative constraints of the surface display postulated by
Kosslyn and others” (Pylyshyn, 1981a, p. 40).

The cognitive penetrability paradigm has also been applied to the mental rota-
tion task (Pylyshyn, 1979b). Pylyshyn reasoned that if mental rotation is accom-
plished by primitive mechanisms, then it must be cognitively impenetrable. One
prediction that follows from this reasoning is that the rate of mental rotation should
be independent of the content being rotated—an image depicting simple content
should, by virtue of its putative architectural nature, be rotated at the same rate as
a different image depicting more complex content.

Pylyshyn (1979b) tested this hypothesis in two experiments and found evidence
of cognitive penetration. The rate of mental rotation was affected by practice, by the
content of the image being rotated, and by the nature of the comparison task that
subjects were asked to perform. As was the case with image scanning, it would seem
that the “analog” rotation of images is not primitive, but is instead based on simpler
processes that do belong to the architecture.

The more carefully we examine phenomena, such as the mental rotation findings,
the more we find that the informally appealing holistic image-manipulation views
must be replaced by finer grained piecemeal procedures that operate upon an ana-
lyzed and structured stimulus using largely serial, resource-limited mechanisms.
(Pylyshyn, 1979b, p. 27)

Cognitive penetrability has played an important role in domains other than mental
imagery. For instance, in the literature concerned with social perception and pre-
diction, there is debate between a classical theory called theory-theory (Gopnik &
Meltzoff, 1997; Gopnik & Wellman, 1992) and a newer approach called simula-
tion theory (Gordon, 1986, 2005b), which is nicely situated in the embodied cog-
nitive science that is the topic of Chapter 5. There is a growing discussion about
whether cognitive penetrability can be used to discriminate between these two theo-
ries (Greenwood, 1999; Heal, 1996; Kuhberger et al., 2006; Perner et al., 1999; Stich
& Nichols, 1997). Cognitive penetrability has also been applied to various topics in
visual perception (Raftopoulos, 2001), including face perception (Bentin & Golland,
2002) and the perception of illusory motion (Dawson, 1991; Dawson & Wright, 1989;
Wright & Dawson, 1994).

While cognitive penetrability is an important tool when faced with the chal-
lenge of examining the architectural equivalence between model and subject, it is
not without its problems. For instance, in spite of it being applied to the study of
mental imagery, the imager debate rages on, suggesting that penetrability evidence
is not as compelling or powerful as its proponents might hope. Perhaps one reason for this is that it seeks a null result—the absence of an effect of cognitive content on cognitive function. While cognitive penetrability can provide architectural evidence for strong equivalence, other sources of evidence are likely required. One source of such additional evidence is cognitive neuroscience.

3.14 Modularity of Mind

Classical cognitive science assumes that cognition is computation, and endorses the physical symbol system hypothesis. As a result, it merges two theoretical positions that in the seventeenth century were thought to be in conflict. The first is Cartesian rationalism, the notion that the products of thought were rational conclusions drawn from the rule-governed manipulation of pre-existing ideas. The second is anti-Cartesian materialism, the notion that the processes of thought are carried out by physical mechanisms.

The merging of rationalism and materialism has resulted in the modification of a third idea, innateness, which is central to both Cartesian philosophy and classical cognitive science. According to Descartes, the contents of some mental states were innate, and served as mental axioms that permitted the derivation of new content (Descartes, 1996, 2006). Variations of this claim can be found in classical cognitive science (Fodor, 1975). However, it is much more typical for classical cognitive science to claim innateness for the mechanisms that manipulate content, instead of claiming it for the content itself. According to classical cognitive science, it is the architecture that is innate.

Innateness is but one property that can serve to constrain theories about the nature of the architecture (Newell, 1990). It is a powerful assumption that leads to particular predictions. If the architecture is innate, then it should be universal (i.e., shared by all humans), and it should develop in a systematic pattern that can be linked to biological development. These implications have guided a tremendous amount of research in linguistics over the last several decades (Jackendoff, 2002). However, innateness is but one constraint, and many radically different architectural proposals might all be consistent with it. What other constraints might be applied to narrow the field of potential architectures?

Another constraining property is modularity (Fodor, 1983). Modularity is the claim that an information processor is not just one homogeneous system used to handle every information processing problem, but is instead a collection of special-purpose processors, each of which is especially suited to deal with a narrower range of more specific problems. Modularity offers a general solution to what is known as the packing problem (Ballard, 1986).
The packing problem is concerned with maximizing the computational power of a physical device with limited resources, such as a brain with a finite number of neurons and synapses. How does one pack the maximal computing power into a finite brain? Ballard (1986) argued that many different subsystems, each designed to deal with a limited range of computations, will be easier to fit into a finite package than will be a single general-purpose device that serves the same purpose as all of the subsystems.

Of course, in order to enable a resource-limited system to solve the same class of problems as a universal machine, a compromise solution to the packing problem may be required. This is exactly the stance adopted by Fodor in his influential 1983 monograph *The Modularity of Mind*. Fodor imagined an information processor that used general central processing, which he called isotropic processes, operating on representations delivered by a set of special-purpose input systems that are now known as modules.

According to Fodor (1983), a module is a neural substrate that is specialized for solving a particular information processing problem. It takes input from transducers, preprocesses this input in a particular way (e.g., computing three-dimensional structure from transduced motion signals [Hildreth, 1983; Ullman, 1979]), and passes the result of this preprocessing on to central processes. Because modules are specialized processors, they are domain specific. Because the task of modules is to inform central processing about the dynamic world, modules operate in a fast, mandatory fashion. In order for modules to be fast, domain-specific, and mandatory devices, they will be “wired in,” meaning that a module will be associated with fixed neural architecture. A further consequence of this is that a module will exhibit characteristic breakdown patterns when its specialized neural circuitry fails. All of these properties entail that a module will exhibit informational encapsulation: it will be unaffected by other models or by higher-level results of isotropic processes. In other words, modules are cognitively impenetrable (Pylyshyn, 1984). Clearly any function that can be shown to be modular in Fodor’s sense must be a component of the architecture.

Fodor (1983) argued that modules should exist for all perceptual modalities, and that there should also be modular processing for language. There is a great deal of evidence in support of this position.

For example, consider visual perception. Evidence from anatomy, physiology, and clinical neuroscience has led many researchers to suggest that there exist
two distinct pathways in the human visual system (Livingstone & Hubel, 1988; Maunsell & Newsome, 1987; Ungerleider & Mishkin, 1982). One is specialised for processing visual form, i.e., detecting an object’s appearance: the “what pathway.” The other is specialised for processing visual motion, i.e., detecting an object’s changing location: the “where pathway.” This evidence suggests that object appearance and object motion are processed by distinct modules. Furthermore, these modules are likely hierarchical, comprising systems of smaller modules. More than 30 distinct visual processing modules, each responsible for processing a very specific kind of information, have been identified (van Essen, Anderson, & Felleman, 1992).

A similar case can be made for the modularity of language. Indeed, the first biological evidence for the localization of brain function was Paul Broca’s presentation of the aphasic patient Tan’s brain to the Paris Société d’Anthropologie in 1861 (Gross, 1998). This patient had profound agrammatism; his brain exhibited clear abnormalities in a region of the frontal lobe now known as Broca’s area. The Chomskyan tradition in linguistics has long argued for the distinct biological existence of a language faculty (Chomsky, 1957, 1965, 1966). The hierarchical nature of this faculty—the notion that it is a system of independent submodules—has been a fruitful avenue of research (Garfield, 1987); the biological nature of this system, and theories about how it evolved, are receiving considerable contemporary attention (Fitch, Hauser, & Chomsky, 2005; Hauser, Chomsky, & Fitch, 2002). Current accounts of neural processing of auditory signals suggest that there are two pathways analogous to the what-where streams in vision, although the distinction between the two is more complex because both are sensitive to speech (Rauschecker & Scott, 2009).

From both Fodor’s (1983) definition of modularity and the vision and language examples briefly mentioned above, it is clear that neuroscience is a key source of evidence about modularity. “The intimate association of modular systems with neural hardwiring is pretty much what you would expect given the assumption that the key to modularity is informational encapsulation” (p. 98). This is why modularity is an important complement to architectural equivalence: it is supported by seeking data from cognitive neuroscience that complements the cognitive penetrability criterion.

The relation between modular processing and evidence from cognitive neuroscience leads us to a controversy that has arisen from Fodor’s (1983) version of modularity. We have listed a number of properties that Fodor argues are true of modules. However, Fodor also argues that these same properties cannot be true of central or isotropic processing. Isotropic processes are not informationally encapsulated, domain specific, fast, mandatory, associated with fixed neural architecture, or cognitively impenetrable. Fodor proceeds to conclude that because isotropic processes do not have these properties, cognitive science will not be able to explain them.
I should like to propose a generalization; one which I fondly hope will someday come to be known as 'Fodor's First Law of the Nonexistence of Cognitive Science.' It goes like this: the more global (e.g., the more isotropic) a cognitive process is, the less anybody understands it. (Fodor, 1983, p. 107)

Fodor's (1983) position that explanations of isotropic processes are impossible poses a strong challenge to a different field of study, called evolutionary psychology (Barkow, Cosmides, & Tooby, 1992), which is controversial in its own right (Stanovich, 2004). Evolutionary psychology attempts to explain how psychological processes arose via evolution. This requires the assumption that these processes provide some survival advantage and are associated with a biological substrate, so that they are subject to natural selection. However, many of the processes of particular interest to evolutionary psychologists involve reasoning, and so would be classified by Fodor as being isotropic. If they are isotropic, and if Fodor’s first law of the nonexistence of cognitive science is true, then evolutionary psychology is not possible.

Evolutionary psychologists have responded to this situation by proposing the massive modularity hypothesis (Carruthers, 2006; Pinker, 1994, 1997), an alternative to Fodor (1983). According to the massive modularity hypothesis, most cognitive processes—including high-level reasoning—are modular. For instance, Pinker (1994, p. 420) has proposed that modular processing underlies intuitive mechanics, intuitive biology, intuitive psychology, and the self-concept. The mind is “a collection of instincts adapted for solving evolutionarily significant problems—the mind as a Swiss Army knife” (p. 420). The massive modularity hypothesis proposes to eliminate isotropic processing from cognition, spawning modern discussions about how modules should be defined and about what kinds of processing are modular or not (Barrett & Kurzban, 2006; Bennett, 1990; Fodor, 2000; Samuels, 1998).

The modern debate about massive modularity indicates that the concept of module is firmly entrenched in cognitive science. The issue in the debate is not the existence of modularity, but is rather modularity’s extent. With this in mind, let us return to the methodological issue at hand, investigating the nature of the architecture. To briefly introduce the types of evidence that can be employed to support claims about modularity, let us consider another topic made controversial by proponents of massive modularity: the modularity of musical cognition.

As we have seen, massive modularity theorists see a pervasive degree of specialization and localization in the cognitive architecture. However, one content area that these theorists have resisted to classify as modular is musical cognition. One reason for this is that evolutionary psychologists are hard pressed to explain how music benefits survival. “As far as biological cause and effect are concerned, music is useless. It shows no signs of design for attaining a goal such as long life, grandchildren, or accurate perception and prediction of the world” (Pinker, 1997, p. 528). As a result, musical processing is instead portrayed as a tangential, nonmodular
function that is inconsequentially related to other modular processes. “Music is auditory cheesecake, an exquisite confection crafted to tickle the sensitive spots of at least six of our mental faculties” (p. 534).

Not surprisingly, researchers interested in studying music have reacted strongly against this position. There is currently a growing literature that provides support for the notion that musical processing—in particular the perception of rhythm and of tonal profile—is indeed modular (Alossa & Castelli, 2009; Peretz, 2009; Peretz & Coltheart, 2003; Peretz & Hyde, 2003; Peretz & Zatorre, 2003, 2005). The types of evidence reported in this literature are good examples of the ways in which cognitive neuroscience can defend claims about modularity.

One class of evidence concerns dissociations that are observed in patients who have had some type of brain injury. In a dissociation, an injury to one region of the brain disrupts one kind of processing but leaves another unaffected, suggesting that the two kinds of processing are separate and are associated with different brain areas. Those who do not believe in the modularity of music tend to see music as being strongly related to language. However, musical processing and language processing have been shown to be dissociated. Vascular damage to the left hemisphere of the Russian composer Shebalin produced severe language deficits but did not affect his ability to continue composing some of his best works (Luria, Tsvetkova, & Futer, 1965). Reciprocal evidence indicates that there is in fact a double dissociation between language and music: bilateral damage to the brain of another patient produced severe problems in music memory and perception but did not affect her language (Peretz et al., 1994).

Another class of evidence is to seek dissociations involving music that are related to congenital brain disorders. Musical savants demonstrate such a dissociation: they exhibit low general intelligence but at the same time demonstrate exceptional musical abilities (Miller, 1989; Pring, Woolf, & Tadic, 2008). Again, the dissociation is double. Approximately 4 percent of the population is tone deaf, suffering from what is called congenital amusia (Ayotte, Peretz, & Hyde, 2002; Peretz et al., 2002). Congenital amusics are musically impaired, but they are of normal intelligence and have normal language abilities. For instance, they have normal spatial abilities (Tillmann et al., 2010), and while they have short-term memory problems for musical stimuli, they have normal short-term memory for verbal materials (Tillmann, Schulze, & Foxton, 2009). Finally, there is evidence that congenital amusia is genetically inherited, which would be a plausible consequence of the modularity of musical processing (Peretz, Cummings, & Dube, 2007).

A third class of evidence that cognitive neuroscience can provide about modularity comes from a variety of techniques that noninvasively measure regional brain activity as information processing occurs (Cabeza & Kingstone, 2006; Gazzaniga, 2000). Brain imaging data can be used to seek dissociations and
attempt to localize function. For instance, by seeing which regions of the brain are active during musical processing but not active when a nonmusical control task is performed, a researcher can attempt to associate musical functions with particular areas of the brain.

Brain imaging techniques have been employed by cognitive neuroscientists interested in studying musical processing (Peretz & Zatorre, 2003). Surprisingly, given the other extensive evidence concerning the dissociation of music, this kind of evidence has not provided as compelling a case for the localization of musical processing in the human brain (Warren, 2008). Instead, it appears to reveal that musical processing invokes activity in many different areas throughout the brain (Schuppert et al., 2000). “The evidence of brain imaging studies has demonstrated that music shares basic brain circuitry with other types of complex sound, and no single brain area can be regarded as exclusively dedicated to music” (Warren, 2008, p. 34). This is perhaps to be expected, under the assumption that “musical cognition” is itself a fairly broad notion, and that it is likely accomplished by a variety of subprocesses, many of which are plausibly modular. Advances in imaging studies of musical cognition may require considering finer distinctions between musical and nonmusical processing, such as studying the areas of the brain involved with singing versus those involved with speech (Peretz, 2009).

Disparities between behavioural evidence concerning dissociations and evidence from brain imaging studies do not necessarily bring the issue of modularity into question. These disparities might simply reveal the complicated relationship between the functional and the implementational nature of an architectural component. For instance, imagine that the cognitive architecture is indeed a production system. An individual production, functionally speaking, is ultra-modular. However, it is possible to create systems in which the modular functions of different productions do not map onto localized physical components, but are instead defined as a constellation of physical properties distributed over many components (Dawson et al., 2000). We consider this issue in a later chapter where the relationship between production systems and connectionist networks is investigated in more detail.

Nevertheless, the importance of using evidence from neuroscience to support claims about modularity cannot be understated. In the absence of such evidence, arguments that some function is modular can be easily undermined.

For instance, Gallistel (1990) has argued that the processing of geometric cues by animals facing the reorientation task is modular in Fodor’s (1983) sense. This is because the processing of geometric cues is mandatory (as evidenced by the pervasiveness of rotational error) and not influenced by “information about surfaces other than their relative positions” (Gallistel, 1990, p. 208). However, a variety of theories that are explicitly nonmodular are capable of generating appropriate rotational error in a variety of conditions (Dawson, Dupuis, & Wilson, 2010; Dawson et al., 2010);
Miller, 2009; Miller & Shettleworth, 2007, 2008; Nolfi, 2002). As a result, the modularity of geometric cue processing is being seriously re-evaluated (Cheng, 2008).

In summary, many researchers agree that the architecture of cognition is modular. A variety of different kinds of evidence can be marshaled to support the claim that some function is modular and therefore part of the architecture. This evidence is different from, and can complement, evidence about cognitive penetrability. Establishing the nature of the architecture is nonetheless challenging and requires combining varieties of evidence from behavioural and cognitive neuroscientific studies.

3.15 Reverse Engineering

Methodologically speaking, what is classical cognitive science? The goal of classical cognitive science is to explain an agent’s cognitive abilities. Given an intact, fully functioning cognitive agent, the classical cognitive scientist must construct a theory of the agent’s internal processes. The working hypothesis is that this theory will take the form of a physical symbol system. Fleshing this hypothesis out will involve proposing a theory, and hopefully a working computer simulation, that will make explicit proposals about the agent’s symbol structures, primitive processes, and system of control.

Given this scenario, a classical cognitive scientist will almost inevitably engage in some form of reverse engineering.

In reverse engineering, one figures out what a machine was designed to do. Reverse engineering is what the boffins at Sony do when a new product is announced by Panasonic, or vice versa. They buy one, bring it back to the lab, take a screwdriver to it, and try to figure out what all the parts are for and how they combine to make the device work. (Pinker, 1997, p. 21)

The reverse engineering conducted by a classical cognitive science is complicated by the fact that one can’t simply take cognitive agents apart with a screwdriver to learn about their design. However, the assumption that the agent is a physical symbol system provides solid guidance and an effective methodology.

The methodology employed by classical cognitive science is called functional analysis (Cummins, 1975, 1983). Functional analysis is a top-down form of reverse engineering that maps nicely onto the multiple levels of investigation that were introduced in Chapter 2.

Functional analysis begins by choosing and defining a function of interest to explain. Defining a function of interest entails an investigation at the computational level. What problem is being solved? Why do we say this problem is being solved and not some other? What constraining properties can be assumed to aid the
solution to the problem? For instance, we saw earlier that a computational theory of language learning (identifying a grammar in the limit) might be used to motivate possible properties that must be true of a language or a language learner.

The next step in a functional analysis is to decompose the function of interest into a set of subcomponents that has three key properties. First, each subcomponent is defined functionally, not physically. Second, each subcomponent is simpler than the original function. Third, the organization of the subcomponents—the flow of information from one component to another—is capable of producing the input-output behaviour of the original function of interest. “Functional analysis consists in analyzing a disposition into a number of less problematic dispositions such that the programmed manifestation of these analyzing dispositions amounts to a manifestation of the analyzed disposition” (Cummins, 1983, p. 28). These properties permit the functional analysis to proceed in such a way that Ryle’s regress will be avoided, and that eventually the homunculi produced by the analysis (i.e., the functional subcomponents) can be discharged, as was discussed in Chapter 2.

The analytic stage of a functional analysis belongs to the algorithmic level of analysis. This is because the organized system of subfunctions produced at this stage is identical to a program or algorithm for producing the overall input-output behaviour of the agent. However, the internal cognitive processes employed by the agent cannot be directly observed. What methods can be used to carve up the agent’s behaviour into an organized set of functions? In other words, how can observations of behaviour support decisions about functional decomposition?

The answer to this question reveals why the analytic stage belongs to the algorithmic level of analysis. It is because the empirical methods of cognitive psychology are designed to motivate and validate functional decompositions.

For example, consider the invention that has become known as the modal model of memory (Baddeley, 1986), which was one of the triumphs of cognitivism in the 1960s (Shiffrin & Atkinson, 1969; Waugh & Norman, 1965). According to this model, to-be-remembered information is initially kept in primary memory, which has a small capacity and short duration, and codes items acoustically. Without additional processing, items will quickly decay from primary memory. However, maintenance rehearsal, in which an item from memory is spoken aloud and thus fed back to the memory in renewed form, will prevent this decay. With additional processing like maintenance rehearsal, some of the items in primary memory pass into secondary memory, which has large capacity and long duration, and employs a semantic code.

The modal memory model was inspired and supported by experimental data. In a standard free-recall experiment, subjects are asked to remember the items from a presented list (Glanzer & Cunitz, 1966; Postman & Phillips, 1965). The first few items presented are better remembered than the items presented in the middle—the primacy effect. Also, the last few items presented are better remembered than
the middle items—the recency effect. Further experiments demonstrated a functional dissociation between the primacy and recency effects: variables that influenced one effect left the other unaffected. For example, introducing a delay before subjects recalled the list eliminated the recency effect but not the primacy effect (Glanzer & Cunitz, 1966). If a list was presented very quickly, or was constructed from low-frequency words, the primacy effect—but not the recency effect—vanished (Glanzer, 1972). To explain such functional dissociation, researchers assumed an organized system of submemories (the modal model), each with different properties.

The analytic stage of a functional analysis is iterative. That is, one can take any of the subfunctions that have resulted from one stage of analysis and decompose it into an organized system of even simpler sub-subfunctions. For instance, as experimental techniques were refined, the 1960s notion of primary memory has been decomposed into an organized set of subfunctions that together produce what is called working memory (Baddeley, 1986, 1990). Working memory is decomposed into three basic subfunctions. The central executive is responsible for operating on symbols stored in buffers, as well as for determining how attention will be allocated across simultaneously ongoing tasks. The visuospatial buffer stores visual information. The phonological loop is used to store verbal (or speech-like) information. The phonological loop has been further decomposed into subfunctions. One is a phonological store that acts as a memory by holding symbols. The other is a rehearsal process that preserves items in the phonological store.

We saw in Chapter 2 that functional decomposition cannot proceed indefinitely if the analysis is to serve as a scientific explanation. Some principles must be applied to stop the decomposition in order to exit Ryle’s regress. For Cummins’ (1983) functional analysis, this occurs with a final stage—causal subsumption. To causally subsume a function is to explain how physical mechanisms bring the function into being. “A functional analysis is complete when the program specifying it is explicable via instantiation—i.e., when we can show how the program is executed by the system whose capacities are being explained” (p. 35). Cummins called seeking such explanations of functions the subsumption strategy. Clearly the subsumption strategy is part of an architectural level of investigation, employing evidence involving cognitive impenetrability and modularity. It also leans heavily on evidence gathered from an implementational investigation (i.e., neuroscience).

From a methodological perspective, classical cognitive science performs reverse engineering, in the form of functional analysis, to develop a theory (and likely a simulation) of cognitive processing. This enterprise involves both formal and empirical methods as well as the multiple levels of investigation described in Chapter 2. At the same time, classical cognitive science will also be involved in collecting data to establish the strong equivalence between the theory and the agent by establishing
links between the two at the different levels of analysis, as we have been discussing in the preceding pages of the current chapter.

3.16 What is Classical Cognitive Science?

The purpose of the current chapter was to introduce the foundations of classical cognitive science—the “flavour” of cognitive science that first emerged in the late 1950s—and the school of thought that still dominates modern cognitive science. The central claim of classical cognitive science is that “cognition is computation.” This short slogan has been unpacked in this chapter to reveal a number of philosophical assumptions, which guide a variety of methodological practices.

The claim that cognition is computation, put in its modern form, is identical to the claim that cognition is information processing. Furthermore, classical cognitive science views such information processing in a particular way: it is processing that is identical to that carried out by a physical symbol system, a device like a modern digital computer. As a result, classical cognitive science adopts the representational theory of mind. It assumes that the mind contains internal representations (i.e., symbolic expressions) that are in turn manipulated by rules or processes that are part of a mental logic or a (programming) language of thought. Further to this, a control mechanism must be proposed to explain how the cognitive system chooses what operation to carry out at any given time.

The classical view of cognition can be described as the merging of two distinct traditions. First, many of its core ideas—appeals to rationalism, computation, innateness—are rooted in Cartesian philosophy. Second, it rejects Cartesian dualism by attempting to provide materialist explanations of representational processing. The merging of rationality and materialism is exemplified by the physical symbol system hypothesis. A consequence of this is that the theories of classical cognitive science are frequently presented in the form of working computer simulations.

In Chapter 2, we saw that the basic properties of information processing systems required that they be explained at multiple levels. Not surprisingly, classical cognitive scientists conduct their business at multiple levels of analysis, using formal methods to answer computational questions, using simulation and behavioural methods to answer algorithmic questions, and using a variety of behavioural and biological methods to answer questions about architecture and implementation.

The multidisciplinary nature of classical cognitive science is revealed in its most typical methodology, a version of reverse engineering called functional analysis. We have seen that the different stages of this type of analysis are strongly related to the multiple levels of investigations that were discussed in Chapter 2. The same relationship to these levels is revealed in the comparative nature of classical cognitive
science as it attempts to establish the strong equivalence between a model and a modelled agent.

The success of classical cognitive science is revealed by its development of successful, powerful theories and models that have been applied to an incredibly broad range of phenomena, from language to problem solving to perception. This chapter has emphasized some of the foundational ideas of classical cognitive science at the expense of detailing its many empirical successes. Fortunately, a variety of excellent surveys exist to provide a more balanced account of classical cognitive science’s practical success (Bechtel, Graham, & Balota, 1998; Bermúdez, 2010; Boden, 2006; Gleitman & Liberman, 1995; Green, 1996; Kosslyn & Osherson, 1995; Lepore & Pylyshyn, 1999; Posner, 1991; Smith & Osherson, 1995; Stillings, 1995; Stillings et al., 1987; Thagard, 1996; Wilson & Keil, 1999).

Nevertheless, classical cognitive science is but one perspective, and it is not without its criticisms and alternatives. Some cognitive scientists have reacted against its avoidance of the implementational (because of multiple realization), its reliance on the structure/process distinction, its hypothesis that cognitive information processing is analogous to that of a digital computer, its requirement of internal representations, and its dependence on the sense-think-act cycle. Chapter 4 turns to the foundations of a different “flavour” of cognitive science that is a reaction against the classical approach: connectionist cognitive science.