# PHILOSOPHY OF MIND

THIRD EDITION

JAEGWON KIM


A Member of the Perseus Books Group

Westview Press was founded in 1975 in Boulder, Colorado, by notable publisher and intellectual Fred Praeger. Westview Press continues to publish scholarly titles and high-quality undergraduate- and graduate-level textbooks in core social science disciplines. With books developed, written, and edited with the needs of serious nonfiction readers, professors, and students in mind, Westview Press honors its long history of publishing books that matter.



Published by Westview Press,
A Member of the Perseus Books Group



Find us on the World Wide Web at www.westviewpress.com.



Westview Press books are available at special discounts for bulk purchases in the United States by corporations, institutions, and other organizations. For more information, please contact the Special Markets Department at the Perseus Books Group, 2300 Chestnut Street, Suite 200, Philadelphia, PA 19103, or call (800) 810-4145, ext. 5000, or e-mail special.markets@perseusbooks.com.

Designed by Trish Wilkinson
Set in 10.5 point Minion Pro

Library of Congress Cataloging-in-Publication Data

Kim, Jaegwon.
Philosophy of mind / Jaegwon Kim.—3rd ed.
p. cm.
ISBN 978-0-8133-4458-4 (alk. paper)
1. Philosophy of mind. I. Title.
BD418.3.K54 2011
128'.2—dc22
E-book ISBN 978-0-8133-4520-8 2010040944

10 9 8 7 6 5 4 3 2 1

CHAPTER 5

# Mind as a Computing Machine

## *Machine Functionalism*

In 1967 Hilary Putnam published a paper of modest length titled "Psychological Predicates."[1] This paper changed the debate in philosophy of mind in a fundamental way, by doing three remarkable things: First, it quickly brought about the decline and fall of type physicalism, in particular, the psychoneural identity theory. Second, it ushered in functionalism, which has since been a highly influential—arguably the dominant—position on the nature of mind. Third, it was instrumental in installing antireductionism as the orthodoxy on the nature of psychological properties. Psychoneural identity physicalism, which had been promoted as the only view of the mind properly informed by the best contemporary science, turned out to be unexpectedly short-lived, and by the mid-1970s most philosophers had abandoned reductionist physicalism not only as a view about psychology but as a doctrine about all special sciences, sciences other than basic physics.[2] In a rapid shift of fortune, identity physicalism was gone in a matter of a few years, and functionalism was quickly enthroned as the "official" philosophy of the burgeoning cognitive science, a view of psychological and cognitive properties that best fit the projects and practices of the scientists.

All this stemmed from a single idea: *the multiple realizability of mental properties*. We have already discussed it as an argument against the psychoneural identity theory and, more generally, as a difficulty for type physicalism (chapter 4). What sets the multiple realization argument apart from numerous other objections to the psychoneural identity theory is that it gave birth to an

attractive new conception of the mental that has played a key role in shaping an influential view of the nature and status of not only cognitive science and psychology but also other special sciences.

## Multiple Realizability and the Functional Conception of Mind

Perhaps not many of us now believe in angels—purely spiritual and immortal beings supposedly with a full mental life. Angels, as traditionally conceived, are wholly immaterial beings with knowledge and belief who can experience emotions and desires and are capable of performing actions. The idea of such a being may be a perfectly coherent one, like the idea of a unicorn or Bigfoot, but there seems no empirical evidence that there are beings fitting the description, just as there are no unicorns and probably no Bigfoot. So like unicorns but unlike married bachelors or four-sided triangles, there seems nothing conceptually impossible about angels. If the idea of an angel with beliefs, desires, and emotions is a consistent one, that would show that there is nothing in the idea of mentality as such that precludes purely nonphysical, wholly immaterial beings with psychological states.[3]

It seems, then, that we cannot set aside the possibility of immaterial realizations of mentality as a matter of an a priori conceptual fact.[4] Ruling out such a possibility requires commitment to a substantive metaphysical thesis, perhaps something like this:

> *Realization Physicalism*. If something *x* has some mental property M (or is in mental state M) at time *t*, then *x* is a physical thing and *x* has M at *t* in virtue of the fact that *x* has at *t* some physical property P that realizes M in *x* at *t*. [5]

It is useful to think of this principle as a way of stating the thesis of physicalism.[6] It says that anything that exhibits mentality must be a physical system—for example, a biological organism. Although the idea of nonphysical entities having mental properties may be a consistent one, the actual world is so constituted, according to this thesis, that only physical systems, like biological organisms, turn out to have mental properties—maybe because they are the only things that exist in space-time. Moreover, the principle requires that every mental property be physically based; each occurrence of a mental property is due to the occurrence of a physical "realizer" of the mental property. A simple way of putting the point would be this: Minds, if they exist, must be embodied.

Notice that this principle provides for the possibility of multiple realization of mental properties. Mental property M—say, being in pain—may be such that in humans C-fiber activation realizes it but in other species (say, octopuses and reptiles) physiological mechanisms that realize pain may be vastly different. Perhaps there might be non-carbon-based or non-protein-based biological organisms with mentality, and we cannot a priori preclude the possibility that electromechanical systems, like the "intelligent" robots and androids in science fiction, might be capable of having beliefs, desires, and even sensations. All this suggests an interesting feature of mental concepts: They seem to carry no constraint on the actual physical-biological mechanisms that realize or implement them. In this sense, psychological concepts are like concepts of artifacts. For example, the idea of an "engine" is silent on how an engine might be designed and built—whether it uses gasoline or electricity or steam and, if it is a gasoline engine, whether it is a piston or rotary engine, how many cylinders it has, whether it uses a carburetor or fuel injection, and so on. As long as a physical device is capable of performing a certain specified job—in this instance, that of transforming various forms of energy into mechanical force or motion—it counts as an engine. The concept of an engine is defined by a *job description*, or *causal role*, not a description of mechanisms that execute the job. Many biological concepts are similar: What makes an organ a heart is the fact that it pumps blood. The human heart may be physically very unlike hearts in, say, reptiles or birds, but they all count as hearts because of the job they do in the organisms in which they are found, not on account of their similarity in shape, size, or material composition.

What, then, is the job description of pain? The capacity for experiencing pain under appropriate conditions—in particular, when an organism suffers tissue damage—is critical to its chances for adaptation and survival. There are unfortunate people who congenitally lack the capacity to sense pain, and few of them survive into adulthood.[7] In the course of coping with the hazards presented by their environment, animal species must have had to develop pain mechanisms, "tissue-damage detectors," and it is plausible that different species, interacting with different environmental conditions and evolving independently, have developed different mechanisms for this purpose. As a start, then, we can think of pain as specified by the job description "tissue-damage detector"—a mechanism that is activated by tissue damage and whose activation in turn causes behavioral responses such as withdrawal, avoidance, and escape.

Thinking of the workings of the mind in analogy with the operations of a computing machine is commonplace, both in the popular press and in serious philosophy and cognitive science, and we will soon begin looking into the

mind-computer analogy in detail. A computational view of mentality also shows that we must expect mental states to be multiply realized. We know that any computational process can be implemented in a variety of physically diverse computing machines. Not only are there innumerable kinds of electronic digital computers (in addition to the semiconductor-based machines we are familiar with, think of the vacuum-tube computers of olden days), but also computers can be built with wheels and gears (as in Charles Babbage's original "Analytical Engine") or even with hydraulically operated systems of pipes and valves, although these would be unacceptably slow (not to say economically prohibitive). And all of these physically diverse computers can be performing "the same computation," say, solving the same differential equations. If minds are like computers and mental processes—in particular, cognitive processes—are, at bottom, computational processes, we should expect no prior constraint on just how minds and mental processes are physically implemented, that is, realized. Just as vastly different physical devices can execute the same computational program, so vastly different biological or physical systems should be able to subserve the same cognitive processes. Such is the core of the functionalist conception of the mind.

What these considerations point to, according to some, is the *abstractness* or *formality* of psychological properties in relation to physical or biological properties: Psychological kinds abstract from the physical and biological details of organisms so that states that are quite unlike from a physicochemical point of view can fall under the same psychological kind, and organisms and systems that are widely dissimilar biologically and physically can instantiate the same psychological regularities—or have "the same psychology." Psychological kinds seem to track *formal* patterns or structures of events and processes rather than their material constitutions or implementing physical mechanisms.[8] Conversely, the same physical structure, depending on the way it is causally embedded in a larger system, can subserve different psychological capacities and functions (just as the same computer chip can be used for different computational functions in the subsystems of a computer). After all, most neurons, it has been observed, are pretty much alike and largely interchangeable.[9]

What is it, then, that binds together all the physically diverse instances of a given mental kind? What do all pains—pains in humans, pains in canines, pains in octopuses, and pains in Martians—have in common in virtue of which they all fall under a single psychological kind, pain?[10] That is, what is the *principle of individuation* for mental kinds?

Let us first see how the type physicalist and the behaviorist answer this question. The psychoneural identity physicalist will say this: What all pains

have in common that makes them instances of pain is a certain neurobiological property, namely, being an instance of C-fiber excitation (or some such state). That is, for the type physicalist, a mental kind is a physical kind (a neurobiological kind, for the psychoneural identity theorist). You could guess how the behaviorist answers the question: What all pains have in common is a certain behavioral property—or to put it another way, two organisms are both in pain at a time just in case at that time they exhibit, or are disposed to exhibit, the behavior patterns characteristic of pain (for example, escape behavior, withdrawal behavior, and so on). For the behaviorist, then, a mental kind is a behavioral kind.

If you take the multiple realizability of mental states seriously, you will reject both these answers and opt for a "functionalist" conception. The main idea is that what is common to instances of a mental state must be sought at a higher level of abstraction. According to functionalism, a mental kind is a *functional kind*, or a *causal-functional kind*, since the "function" involved is to fill a certain causal role.[11] Let us go back to pain as a tissue-damage detector.[12] The concept of a tissue-damage detector is a *functional concept*, a concept specified by a job description, as we said: Any device is a tissue-damage detector for an organism just in case it can reliably respond to occurrences of damage to the tissues of the organism and transmit this information to other subsystems so that appropriate responses are produced. Functional concepts are ubiquitous: What makes something a mousetrap, a carburetor, or a thermometer is its ability to perform a certain function, not any specific physicochemical structure or mechanism; as someone said, anything is a mousetrap if it takes a live mouse as input and delivers a dead one as output. These concepts are specified by the functions that are to be performed, not by structural blueprints. As has been noted, many concepts, in ordinary discourse and in the sciences, are functional concepts in this sense; important concepts in chemistry and biology (for example, catalyst, gene, heart) seem best understood as functional concepts.

To return to pain as a tissue-damage detector: Ideally, every instance of tissue damage, and nothing else, should activate this mechanism and this must further trigger other mechanisms with which it is hooked up, leading finally to behavior that will in normal circumstances spatially separate the damaged part, or the whole organism, from the external cause of the damage. Thus, the concept of pain is defined in terms of its function, and the function involved is to serve as a *causal intermediary* between typical pain inputs (tissue damage, trauma, and so on) and typical pain outputs (winces, groans, avoidance behavior, and so on). Moreover, functionalism makes two significant additions. First, the causal conditions that activate the pain mechanism can include other

mental states (for example, you must be normally alert and not be absorbed in another activity, like intense competitive sports). Second, the outputs of the pain mechanism can include mental states as well (such as a sense of distress or a desire to be rid of the pain). Mental kinds are causal-functional kinds, and what all instances of a given mental kind have in common is that they all serve a certain *causal role* distinctive of that kind. And that is all. One might say that a functional kind has only a "nominal essence," given by its defining causal role, but no "real essence," a "deep" common property shared by all actual and possible instances of it.[13] Contrast this with water: All samples of water, anywhere anytime, must be quantities of $H_2O$ molecules, and being composed of $H_2O$ molecules is the essence of water. Pain does not have an essence in that sense. Functionalism itself may be characterized by the following slogan: "Psychological kinds have only nominal essences; they have no real essences."

In general, then, as David Armstrong has put it, the concept of a mental state is the concept of an internal state apt to be caused by certain sensory inputs and apt to cause certain behavioral outputs. A specification of input and output, <i, o>, will define a particular mental state: for example, <tissue damage, aversive behavior> defines pain, <skin irritation, scratching> defines itch, and so on.

## Functional Properties and Their Realizers: Definitions

It will be useful to have explicit definitions of some of the terms we have been using informally, relying on examples and intuitions. Let us begin with a more precise characterization of a functional property:

> F is a *functional property* (or kind) just in case F can be characterized by a definition of the following form:
>
> For something *x* to have F (or to be an F) = $_{\text{def}}$ for *x* to have some property P such that C(P), where C(P) is a specification of the causal work that P is supposed to do in *x*.

We may call a definition having this form a "functional" definition. "C(P)," which specifies the causal role of F, is crucial. What makes a functional property the property it is, is the causal role associated with it; that is to say, F and G are the same functional property if and only if the causal role associated with F is the same as that associated with G. The term "causal work" in the

above schema of functional definitions should be understood broadly to refer to "passive" as well as "active" work: For example, if tissue damage causes P to instantiate in an organism, that is part of P's causal work or function. Thus, P's causal work refers to the *causal relations* involving the instances, or occurrences, of P in the organism or system in question.

Now we can define what it is for a property to "realize," or be a "realizer" of, a functional property:

> Let F be a functional property defined by a functional definition, as above. Property Q is said to *realize* F, or be a *realizer* or a *realization* of F, in system *x* if and only if C(Q), that is, Q fits the specification C in *x* (which is to say, Q in fact performs the specified causal work in system *x*).

Note that the definiens (the right-hand side) of a functional definition does not mention any particular property P that *x* has (when it has F); it only says that *x* has "some" property P fitting description C. In logical terminology, the definiens "existentially quantifies over" properties (it in effect says, "There exists some property P such that *x* has P and C[P]"). For this reason, functional properties are called "second-order" properties, with the properties quantified over (that is, properties eligible as instances of P) counting as "first-order" properties; they are second-order properties of a special kind—namely, those that are defined in terms of causal roles.

Let us see how this formal apparatus works. Consider the property of being a mousetrap. It is a functional property because it can be given the following functional definition:

> *x* is a mousetrap $=_{\text{def}}$ *x* has some property P such that P enables *x* to trap and hold or kill mice.

The definition does not specify any specific P that *x* must have; the causal work specified obviously can be done in many different ways. There are the familiar spring-loaded traps, and there are wire cages with a door that slams shut when a mouse enters; we can imagine high-tech traps with an optical sensor and all sorts of other devices. This means that there are many—in fact, indefinitely many—"realizers" of the property of being a mousetrap; that is, all sorts of physical mechanisms can be mousetraps.[14] The situation is the same with pain: A variety of physical/biological mechanisms can serve as tissue-damage detectors across biological species—and perhaps nonbiological systems as well.

## Functionalism and Behaviorism

Both functionalism and behaviorism speak of sensory input and behavioral output—or "stimulus" and "response"—as central to the concept of mentality. In this respect, functionalism is part of a broadly behavioral approach to mentality and can be considered a generalized and more sophisticated version of behaviorism. But there are also significant differences between them, of which the following two are the most important.

First, the functionalist takes mental states to be *real internal* states of an organism with causal powers; for an organism to be in pain is for it to be in an internal state (for example, a neurobiological state for humans) that is typically caused by tissue damage and that in turn typically causes winces, groans, and avoidance behavior. And the presence of this internal state explains why humans react the way they do when they suffer tissue damage. In contrast, the behaviorist eschews talk of internal states entirely, identifying mental states with actual or possible behavior. Thus, to be in pain, for the behaviorist, is to wince and groan or be disposed to wince and groan, but not, as the functionalist would have it, to be in some *internal state that causes* winces and groans.

Although both the behaviorist and the functionalist may refer to "behavioral dispositions" in speaking of mental states, what they mean by "disposition" can be quite different: The functionalist takes a "realist" approach to dispositions, whereas the behaviorist embraces an "instrumentalist" line. We say that sugar cubes, for example, are soluble in water. But what does it mean to say that something is soluble in water? The answer depends on whether you adopt an instrumental or a realist view of dispositions. Let us see exactly how these two approaches differ:

> *Instrumentalist analysis*: $x$ is soluble in water $=_{\text{def}}$ if $x$ is immersed in water, $x$ dissolves.

> *Realist analysis*: $x$ is soluble in water $=_{\text{def}}$ $x$ has an internal state S (for example, a certain microstructure) such that when $x$ is immersed in water, S causes $x$ to dissolve.

According to instrumentalism, therefore, all there is to the water solubility of a sugar is the fact that a certain conditional ("if-then") statement holds for it; thus, on this view, water solubility is a "conditional" or "hypothetical" property of the sugar cube—that is, the property of *dissolving if immersed in water*. Realism, in contrast, takes solubility to be a categorical, presumably microstruc-

tural, internal state of the cube of sugar that is causally responsible for its dissolving when placed in water. (Further investigation might reveal the state to be that of having a certain crystalline molecular structure.) Neither analysis requires the sugar cube to be placed in water or actually to be dissolving in order to be water-soluble. However, we may note the following difference: If *x* dissolves in water and *y* does not, the realist will give a causal explanation of this difference in terms of a difference in their microstructure. For the instrumentalist, the difference may just be a brute fact: It is just that the conditional "if placed in water, it dissolves" holds true for *x* but not for *y*, a difference that need not be grounded in any further differences between *x* and *y*.

In speaking of mental states as behavioral dispositions, then, the functionalist takes them as actual inner states of persons and other organisms that in normal circumstances cause behavior of some specific type under certain specified input conditions. Mental states serve as causal intermediaries between sensory input and behavioral output. In contrast, the behaviorist takes mental states merely as input-output, or stimulus-response, correlations. Many behaviorists (especially "radical" scientific behaviorists) believe that speaking of mental states as "inner causes" of behavior is scientifically unmotivated and philosophically unwarranted.[15]

The second significant difference between functionalism and behaviorism, one that gives the former a substantially greater theoretical power, is the way "input" and "output" are construed for mental states. For the behaviorist, input and output consist entirely of observable physical stimulus conditions and observable behavioral/physical responses. As mentioned earlier, the functionalist allows reference to other *mental states* in the characterization of a given mental state. It is a crucial part of the functionalist conception of mental states that their typical causes and effects can, and often do, include other mental states. Thus, for a ham sandwich to cause you to want to eat it, you must *believe* it to be a ham sandwich; a bad headache can cause you not only to frown and moan but also to experience further mental states like *distress* and a *desire* to call your doctor.

The two points that have just been reviewed are related: If you think of mental states as actual inner states of psychological subjects, you would regard them as having real causal powers, powers to cause and be caused by other states and events, and there is no obvious reason to exclude mental states from figuring among the causes or effects of other mental states. In conceiving mentality this way, the functionalist is espousing *mental realism*—a position that considers mental states as having a genuine ontological status and counts them among the phenomena of the world with a place in its causal structure. Mental

states are real for the behaviorist too, but only as behaviors or behavioral dispositions; for him, there is nothing mental over and above actual and possible behavior. For the functionalist, mental states are inner causes of behavior, and as such they are "over and above" behavior.

Including other mental events among the causes and effects of a given mental state is part of the functionalist's general conception of mental states as forming a complex causal network anchored to the external world at various points. At these points of contact, a psychological subject interacts with the outside world, receiving sensory inputs and emitting behavior outputs. And the identity of a given mental kind, whether it is a sensation like pain or a belief that it is going to rain or a desire for a ham sandwich, depends solely on the place it occupies in the causal network. That is, what makes a mental event the kind of mental event it is, is the way it is causally linked to other mental-event kinds and input-output conditions. Since each of these other mental-event kinds in turn has its identity determined by *its* causal relations to other mental events and to inputs and outputs, the identity of each mental kind depends ultimately on the whole system—its internal structure and the way it is causally linked to the external world via sensory inputs and behavior outputs. In this sense, functionalism gives us a *holistic* conception of mentality.

This holistic approach enables functionalism to sidestep one of the principal objections to behaviorism. This is the difficulty we saw earlier: A desire issues in overt behavior only when combined with an appropriate belief, and similarly, a belief leads to behavior only when a matching desire is present. For example, a person with a desire to eat an apple will eat an apple that is presented to her only if she believes it to be an apple (she would not bite into it if she thought it was a fake wooden apple); a person who believes that it is going to rain will take an umbrella only if she has a desire to stay dry. As we saw, this apparently makes it impossible to give a behavioral definition of desire without reference to belief or a definition of belief without reference to desire. The functionalist would say that this simply points to the holistic character of mental states: It is an essential feature of a desire that it is the kind of internal state that in concert with an appropriate belief causes a certain behavior output, and similarly for belief and other mental states.

But doesn't this make the definitions circular? If the concept of desire cannot be defined without reference to belief, and the concept of belief in turn cannot be explained without reference to desire, how can either be understood at all? We will see later (chapter 6) how the holistic approach of functionalism deals with this issue.[16]

## Turing Machines

Functionalism was originally formulated by Putnam in terms of "Turing machines," mathematically characterized computing machines due to the British mathematician-logician Alan M. Turing.[17] Although it is now customary to formulate functionalism in terms of causal-functional roles—as we have done and will do in more detail in the next chapter—it is instructive to begin our systematic treatment of functionalism by examining the Turing-machine version of functionalism, usually called machine functionalism. This also gives us a background that will be helpful in exploring the idea that the workings of the mind are best understood in terms of the operations of a computing machine—that is, the computational view of the mind (computationalism, for short).

A Turing machine is made up of four components:

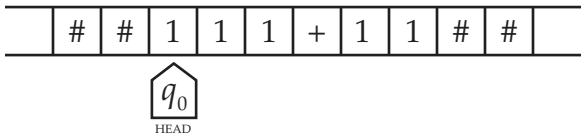
1. A *tape* divided into "squares" and unbounded in both directions
2. A *scanner-printer* ("head") positioned at one of the squares of the tape at any given time
3. A finite set of *internal states* (or *configurations*), $q_0, \ldots, q_n$
4. A finite *alphabet* consisting of symbols, $b_1, \ldots, b_m$

One and only one symbol appears on each square. (We may think of the blank as one of the symbols.)

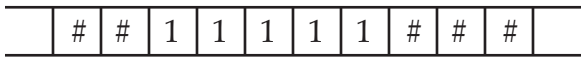
The machine operates in accordance with the following general rules:

A. At each time, the machine is in one of its internal states, $q_i$, and its head is scanning a particular square on the tape.
B. What the machine does at a given time $t$ is completely determined by its internal state at $t$ and the symbol its head is scanning at $t$.
C. Depending on its internal state and the symbol being scanned, the machine does three things:
   (1) Its head replaces the symbol with another (possibly the same) symbol of the alphabet. (To put it another way, the head erases the symbol being scanned and prints a new one, which may be the same as the erased one.)
   (2) Its head moves one square to the right or to the left (or halts, with the computation completed).
   (3) The machine enters into one of its internal states (which can be the same state).

Let us consider a Turing machine that adds positive integers in the unary notation. (In this notation, number $n$ is represented as a sequence of $n$ strokes, each stroke occupying one square.) Consider the following picture in which the problem of adding 3 and 2 is presented to the machine, which is to be started off with its head in state $q_0$ and scanning the first digit:




(The "scratch" symbol, #, marks the boundaries of the problem.) We want to "program" this Turing machine in such a way that when the computation is completed, the machine halts with a sequence of five consecutive strokes showing on the tape, like this:




It is easy to see that there are various procedures by which the machine could accomplish this. One simple way is to have the machine (or its head) move to the right looking for the symbol +, replace it with a stroke, keep moving right until it finds the right-most stroke, and when it does, erase it (that is, replace it with the scratch symbol #) and then halt. The following simple "machine table" is a complete set of instructions that defines our adder (call it $TM_1$):

| | $q_0$ | $q_1$ |
|---|---|---|
| 1 | $1Rq_0$ | #Halt |
| + | $1Rq_0$ | |
| # | $\#Lq_1$ | |

Here is how we read this table. On the left-most column you find the symbols of the machine alphabet listed vertically, and the top row lists the machine's internal states. Each entry in the interior matrix is an *instruction*: It tells the machine what to do when it is scanning the symbol shown in the left-most column of that row and is in the internal state listed at the top of the column. For example, the entry $1Rq_0$, at the intersection of $q_0$ and 1, tells the machine: "If you are in internal state $q_0$ and scanning the symbol 1, replace 1 with 1 (that is, leave it unchanged), move to the right by one square, and go into internal state $q_0$ (that is, stay in the same state)." The entry immediately below, $1Rq_0$, tells the machine: "If you are in state $q_0$ and scanning the symbol +, replace + with 1, move to the

right by one square, and go into state $q_0$." The *L* in the bottom entry, $\#Lq_1$, means "move left by one square"; the entry in the right-most column, #Halt, means "If you are scanning 1 and in state $q_1$, replace 1 with # and halt." It is easy to see (the reader is asked to figure this out on her own) the exact sequence of steps our Turing machine will follow to compute the sum 3 + 2.

The machine table of a Turing machine is a complete and exhaustive specification of the machine's operations. We may therefore identify a Turing machine with its machine table. Since a machine table is nothing but a set of instructions, this means that a Turing machine can be identified with a set of such instructions.

What sort of things are the "internal states" of a Turing machine? We talk about this general question later, but with our machine $TM_1$, it can be helpful to think of the specific machine states in the following intuitive way: $q_0$ is a + and # searching state—it is a state such that when $TM_1$ is in it, it keeps going right, looking for + and #, ignoring any 1s it encounters. Moreover, if the machine is in $q_0$ and finds a +, it replaces it with a 1 and keeps moving to the right, while staying in the same state; when it scans a # (thereby recognizing the right-most boundary of the given problem), it backs up to the left and goes into a new state $q_1$, the "print # over 1 and then halt" state. When $TM_1$ is in this state, it will replace any 1 it scans with a # and halt. Thus, each state "disposes" the machine to do a set of specific things depending on the symbol being scanned (which therefore can be likened to sensory input).

But this is not the only Turing machine that can add numbers in unary notation; there is another one that is simpler and works faster. It is clear that to add unary numbers it is not necessary for the machine to determine the right-most boundary of the given problem; all it needs to do is to erase the initial 1 being scanned when it is started off, and then move to the right to find + and replace it with a 1. This is $TM_2$, with the following machine table:

| | $q_0$ | $q_1$ |
|---|---|---|
| 1 | $\#Rq_1$ | $1Rq_1$ |
| + | | 1Halt |
| # | | |

We can readily build a third Turing machine, $TM_3$, that will do subtractions in the unary notation. Suppose the following subtraction problem is presented to the machine:

| # | # | *b* | 1 | 1 | 1 | 1 | – | 1 | 1 | *b* | # | # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

(Symbol *b* is used to mark the boundaries of the problem.) Starting the machine in state $q_0$ scanning the initial 1, we can write a machine table that computes *n*–*m* by operating like this:

1. The machine starts off scanning the first 1 of *n*. It goes to the right until it locates *m*, the number being subtracted. (How does it recognize it has located *m*?) It then erases the first 1 of this number (replacing it with a #), goes left, and erases the last 1 of *n* (again replacing it with a #).
2. The machine then goes right and repeats step 1 again and again, until it exhausts all the 1s in *m*. (How does the machine "know" that it has done this?) We then have the machine move right until it locates the subtraction sign–, which it erases (that is, replaces it with a #), and then halt. (If you like tidy output tapes, you may have the machine erase the *b*s before halting.)
3. If the machine runs out of the first set of strokes before it exhausts the second set (this means that $n < m$), we can have the machine print a certain symbol, say ?, to mean that the given problem is not well-defined. We must also provide for the case where $n = m$.

The reader is invited to write out a machine table that implements these operations.

We can also think of a "transcription machine," $TM_4$, that transcribes a given string of 1s to its right (or left). That is, if $TM_4$ is presented with the following tape to begin its computation,

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | # | 1 | 1 | 1 | # | # | # | # | # | # | # | |

it ends with the following configuration of symbols on its tape:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | # | 1 | 1 | 1 | # | 1 | 1 | 1 | # | # | # | # |

The interest of the transcription machine lies in how it can be used to construct a multiplication machine, $TM_5$. The basic idea is simple: We can get $n \times m$ by transcribing the string of *n* 1s *m* times (that is, transcribing *n* repeatedly using *m* as a counter). The reader is encouraged to write a machine table for $TM_5$.

Since any arithmetical operation (squaring, taking the factorial, and so on) on natural numbers can be defined in terms of addition and multiplication,

it follows that there is a Turing machine that computes any arithmetical operation. More generally, it can be shown that any computation performed by any computer can be done by a Turing machine. That is, being computable and being computable by a Turing machine turn out to be equivalent.[18] In this sense, the Turing machine captures the general idea of computation and computability.

We can think of a Turing machine with two separate tapes (one for input, on which the problem to be computed is presented, and the other for actual computation and the final output) and two separate heads (one for scanning and one for printing). This helps us to think of a Turing machine as receiving "sensory stimuli" (the symbols on the input tape) through its scanner ("sense organ") and emitting specific behaviors in response (the symbols printed on the output tape by its printer head). It can be shown that any computation that can be done by a two-tape machine or a machine with any finite number of tapes can be done by a one-tape machine. So adding more tapes does not strengthen the computing power of Turing machines or substantively enrich the concept of a Turing machine, although it could speed up computations.

Turing also showed how to build a "universal machine," which is like a general-purpose computer in that it is not dedicated to the computation of a specific function but can be programmed to compute any function you want. On the input tape of this machine, you specify two things: the machine table of the desired function in some standard notation that can be read by the universal machine and the values for which the function is to be computed. The universal machine is programmed to read any machine table and carry out the computation in accordance with the instructions of the machine table.

The notion of a Turing machine can be generalized to yield the notion of a *probabilistic automaton*. As you recall, each instruction of a Turing machine is *deterministic*: Given the internal state and the symbol being scanned, the immediate next operation is wholly and uniquely determined. An instruction of a probabilistic, or stochastic, automaton has the following general form: Given internal state *qi* and scanned symbol *bj*:

1. Print $b_k$ with probability $r_1$, or print $b_l$ with probability $r_2$, . . . , or print $b_m$ with probability $r_n$ (where the probabilities add up to 1).
2. Move *R* with probability $r_1$, or move *L* with probability $r_2$ (where the probabilities add up to 1).
3. Go into internal state $q_j$ with probability $r_1$, or into $q_k$ with probability $r_2$, . . . , or into $q_m$ with probability $r_n$ (again, the probabilities adding up to 1).

Although in theory a machine can be made probabilistic along any one or more of these three dimensions, it is customary to understand a probabilistic machine as one that incorporates probabilities into state transitions, in the manner of (3) above. The operations of a probabilistic automaton are not deterministic; the current internal state of the machine and the symbol it is scanning do not—do not always, at any rate—together uniquely determine what the machine will do next. However, the behavior of such a machine is not random or arbitrary either: There are fixed and stable probabilities describing the machine's operations. If we are thinking of a machine that describes the behavior of an actual psychological subject, a probabilistic machine may be more realistic than a deterministic one; however, we may note that it is generally possible to construct a deterministic machine that simulates the behavior of a probabilistic machine to any desired degree of accuracy, which makes probabilistic machines theoretically dispensable.

## Physical Realizers of Turing Machines

Suppose that we give the machine table for our simple adding machine, $TM_1$, to an engineering class as an assignment: Each student is to build an actual physical device that will do the computations as specified by its machine table. What we are asking the students to build, therefore, are "physical realizers" of $TM_1$—real-life physical computing machines that will operate in accordance with the machine table of $TM_1$. We can safely predict that a huge, heterogeneous variety of machines will be turned in. Some of them may really look and work like the Turing machine as described: They will have a paper tape neatly divided into squares, with an actual physical "head" that can read, erase, and print symbols. Some will perhaps use magnetic tapes and heads that read, write, and erase electrically. Some machines will have no "tapes" or "heads" but instead use spaces on a computer disk or memory locations in their CPU to do the computation. A clever student with a sense of humor (and lots of time and other resources) might try to build a hydraulically operated device with pipes and valves instead of wires and switches. The possibilities are endless.

But what exactly is a physical realizer of a Turing machine? What makes a physical device a *realizer of* a given Turing machine? First, the symbols of the machine's alphabet must be given concrete physical embodiments; they could be blotches of ink on paper, patterns of magnetized iron particles on plastic tape, electric charges in capacitors, or what have you. Whatever they are, the physical device that does the "scanning" must be able to "read" them—that is,

differentially respond to them—with a high degree of reliability. This means that the physical properties of the symbols place a set of constraints on the physical design of the scanner, but these constraints need not, and usually will not, determine a unique design; a great multitude of physical devices are likely to be adequate to serve as a scanner for any set of physically embodied symbols. The same considerations apply to the machine's printer and outputs as well: The symbols the machine prints on its output tape (we are thinking of a two-tape machine) must be given physical shapes, and the printer must be designed to produce them on demand. The printer, of course, does not have to "print" anything in a literal sense; the operation could be wholly electronic, or the printer could be a speaker that vocalizes the output or an LCD monitor that visually displays it (and saves it for future computational purposes).

What about the "internal states" of the machine? How are they physically realized? Consider a particular instruction on the machine table of $TM_1$: If the machine is in state $q_0$ and scanning a +, replace the + with a 1, move right, and go into state $q_1$. Assume that $Q_0$ and $Q_1$ are the physical states realizing $q_0$ and $q_1$, respectively. $Q_0$ and $Q_1$, then, must satisfy the following condition: An occurrence of $Q_0$, together with the physical scanning of +, must *physically cause* three physical events: (1) The physical symbol + is replaced with the physical symbol 1; (2) the physical scanner-printer (head) moves one square to the right (on the physical tape) and scans it; and (3) the machine enters state $Q_1$. In general, then, what needs to be done is to *replace the functional or computational relations* among the various abstract parameters (symbols, states, and motions of the head) mentioned in the machine table *with matching causal relations among the physical embodiments* of these parameters. That is to say, a physical realizer of a Turing machine is a physical causal mechanism that is isomorphic to the machine table of the Turing machine.

From the logical point of view, the internal states are only "implicitly defined" in terms of their relations to other parameters: $q_j$ is a state such that if the machine is in it and scanning symbol $b_k$, the machine replaces $b_k$ with $b_l$, moves R (that is, to the right), and goes into state $q_h$; if the machine is scanning $b_m$, it does such and such; and so on. So $q_j$ can be thought of as a function that maps symbols of the alphabet to the triples of the form $<b_k$, R (or L), $q_h>$. From the physical standpoint, $Q_j$, which realizes $q_j$, can be thought of as a *causal* intermediary between the physically realized symbols and the physical realizers of the triples—or equivalently, as a *disposition* to emit appropriate physical outputs (the triples) in response to different physical stimuli (the physical symbols scanned). This means that the intrinsic physical natures of the Qs that realize the *q*s are of no interest to us as long as they have the right

causal powers or capacities; their intrinsic properties do not matter—or more accurately, they matter only to the extent that they affect the desired causal powers of the states and objects that have them. As long as these states perform their assigned causal work, they can be anything you please. Clearly, whether the Qs realize the *q*s depends crucially on how the tape, symbols, and so on are physically realized; in fact, these are interdependent questions. It is plausible to suppose that, with some mechanical ingenuity, a machine could be rewired so that physical states realizing distinct machine states could be interchanged without affecting the operation of the machine.

We see, then, a convergence of two ideas: the functionalist conception of a mental state as a state occupying a certain specific causal role and the idea of a physical state realizing an internal state of a Turing machine. Just as, on the functionalist view, what makes a given mental state the kind of mental state it is, is its causal role with respect to sensory inputs, behavior outputs, and other mental states, so what makes a physical state the realizer of a given internal machine state is its causal relations to inputs, outputs, and other physical realizers of the machine's internal states. This is why it is natural for functionalists to look to Turing machines for a model of the mind.

Let S be a physical system (which may be an electromechanical device like a computer, a biological organism, an auto assembly plant, or anything else), and assume that we have adopted a vocabulary to describe its inputs and outputs. That is, we have a specification of what is to count as the inputs it receives from its surroundings and what is to count as its behavioral outputs. Assume, moreover, that we have specified what states of S are to count as its "internal states." We will say that a Turing machine M is a *machine description* of system S, relative to a given input-output specification and a specification of the internal states, just in case S realizes M relative to the input-output and internal state specifications. Thus, the relation of *being a machine description of* is the converse of the relation of *being a realizer (or realization) of*. We can also define a concept that is weaker than machine description: Let us say that a Turing machine M is a *behavioral description* of S (relative to an input-output specification) just in case M provides a correct description of S's input-output correlations. Thus, every machine description of S is also a behavioral description of S, but the converse does not in general hold. M can give a true description of the input-output relations characterizing S, but its machine states may not be realized in S, and S's inner workings (that is, its computational processes) may not correctly mirror the functional-computational relationships given by M's machine table. In fact, there may be another Turing machine M*, distinct from M, that gives a correct machine description of S. It follows, then,

that *two physical systems that are input-output equivalent may not be realizations of the same Turing machine*. (The pair of adding machines $TM_1$ and $TM_2$ is a simple example of this.)

## Machine Functionalism: Motivations and Claims

Machine functionalists claim that we can think of the mind as a Turing machine (or a probabilistic automaton). This of course needs to be filled out, but from the preceding discussion it should be pretty clear how the story will go. The central idea is that what it is for something to have mentality—that is, to have a psychology—is for it to be a physically realized Turing machine of appropriate complexity, with its mental states (that is, mental-state types) identified with the realizers of the internal states of the machine table. Another way of explaining this idea is to use the notion of machine description: An organism has mentality just in case there is a Turing machine of appropriate complexity that is a machine description of it, and its mental-state kinds are to be identified with the physically realized internal states of that Turing machine. All this is, of course, relative to an appropriately chosen input-output specification, since you must know, or decide, what is to count as the organism's inputs and outputs before you can determine what Turing machine (or machines) it can be said to realize.

Let us consider the idea that *the psychology of an organism* can be represented by a Turing machine, an idea that is commonly held by machine functionalists.[19] Let V be a complete specification of all possible inputs and outputs of a psychological subject S, and let C be all actual and possible input-output correlations of S (that is, C is a complete specification of which input applied to S elicits which output, for all inputs and outputs listed in V). In constructing a *psychology* for S, we are trying to formulate a *theory* that gives a perspicuous systematization of C by positing a set of internal states in S. Such a theory *predicts* for any input applied to S what output will be emitted by S and also *explains* why that particular input will elicit that particular output. It is reasonable to suppose that for any behavioral system complex enough to have a psychology, this kind of systematization is not possible unless we advert to its internal states, for we must expect that the same input applied to S does not always prompt S to produce the same output. The actual output elicited by a given input depends, we must suppose, on the internal state of S at that time.

Before we proceed further, it is necessary to modify our notion of a Turing machine in one respect: The internal states, *q*s, of a Turing machine are *total*

states of the machine at a given time, and the Qs that are their physical realizers are also *total* physical states at a time of the physically realized machine. This means that the Turing machines we are talking about are not going to look very much like the psychological theories we are familiar with; the states posited by these theories are seldom, if ever, total states of a subject at a time. But this is a technical problem, something we assume can be remedied with a more fine-grained notion of an "internal state." We can then think of a total internal state as made up of these "partial" states, which combine in different ways to yield different total states. This modification should not change anything essential in the original conception of a Turing machine. In the discussion to follow, we use this modified notion of an internal state in most contexts.

To return to the question of representing the psychology of a subject S in terms of a Turing machine: What Turing machine, or machines, is adequate as a description of S's psychology? Evidently, any adequate Turing machine must be a behavioral description of S, in the sense defined earlier; that is, it must give a correct description of S's input-output relations (relative to V). But as we have seen, there is bound to be more than one Turing machine—in fact, if there is one, there will be indefinitely more—that gives a correct representation of S's input-output relations.

Since each of these machines is a correct behavioral description of our psychological subject S, they are all equally good as *predictive* theories: Although some of them may be easier to manipulate and computationally more efficient than others, they all predict the same behavior output for the same input. This is a simple consequence of the notion of "behavioral description." But they are different as Turing machines. But do the differences between them matter?

It should be clear how behaviorally equivalent Turing machines, say, $M_1$ and $M_2$, can differ from each other. To say that they are different Turing machines is to say that their machine tables are different—that is how Turing machines are individuated. This means that when they are given the same input, $M_1$ and $M_2$ are likely to go through *different computational processes* to arrive at the same output. Each machine has a set of internal states—let us say $<q_0, q_1, \ldots, q_n>$ for $M_1$ and $<r_0, r_1, \ldots, r_m>$ for $M_2$. Let us suppose further that $M_1$ is a machine description of our psychological subject S, but $M_2$ is not. That is, S is a physical realizer of $M_1$ but not of $M_2$. This means that the computational relations represented in $M_1$, but not those represented in $M_2$, are mirrored in a set of causal relations among the physical-psychological states of S. So there are real physical (perhaps neurobiological) states in S, $<Q_0, Q_1, \ldots, Q_n>$, corresponding to $M_1$'s internal states $<q_0, q_1, \ldots, q_n>$, and these

Qs are causally hooked up to each other and to the physical scanner (sense organs) and the physical printer (motor mechanisms) in a way that ensures that for all computational processes generated by $M_1$, isomorphic causal processes occur in S. As we may say, S is a "causal isomorph" of $M_1$.

There is, then, a clear sense in which $M_1$ is, but $M_2$ is not, *psychologically real* for S, even though they are both accurate predictive theories of S's observable input-output behaviors. $M_1$ gives "the true psychology" of S in that, as we saw, S has a physical structure whose states constitute a causal system that mirrors the computational structure represented by the machine table of $M_1$, and the physical-causal operations of S form an isomorphic image of the computational operations of $M_1$. This makes a crucial difference when what we want is an *explanatory* theory, a theory that *explains why, and how, S does what it does under the given input conditions*. Suppose we say: When input *i* was applied to S, S emitted behavioral output *o because* it was in internal state Q. This can count as an explanation, it seems, only if the state appealed to—namely, Q—is a "real" state of the system. In particular, it can count as a *causal* explanation only if the state Q is what, in conjunction with *i*, caused *o*. Since S is a physical realizer of $M_1$, or equivalently, $M_1$ is a machine description of S, the causal process leading from Q and input *i* to behavior output *o* is mirrored exactly by the computational process that occurs in accordance with the machine table of $M_1$. In contrast, Turing machine $M_2$, which is not realized by S, has no "inner" psychological reality for S, even though it correctly captures all of S's input-output connections. Although, like $M_1$, $M_2$ correlates input *i* with output *o*, the computational process whereby the correlation is effected does not reflect the actual causal process in S that leads from *i* to *o* (or physical embodiments thereof). The explanatory force of "because" in "S emitted *o* when it received input *i because* it was in state Q" derives from the causal relations involving Q and the physical embodiments of *o* and *i* in the system S.

The philosophical issues here depend, partly but critically, on the metaphysics of scientific theories you accept. If you think of scientific theories in general, or theories over some specific domain, merely as predictive instruments that enable us to infer or calculate further observations from the given data, you need not attach any existential significance to the posits of these theories—like the unobservable microparticles of theoretical physics and their (often quite strange) properties—and may regard them only as calculational aids in deriving predictions. A position like this is called "instrumentalism," or "antirealism," about scientific theory.[20] On such a view, the issue of "truth" does not arise for the theoretical principles, nor does the issue of "reality" for the entities and properties posited; the only thing that matters is the

"empirical, or predictive, adequacy" of the theory—how accurately the theory works as a predictive device and how comprehensive its coverage is. If you accept an instrumentalist stance toward psychological theory, therefore, any Turing machine that is a behavioral description of a psychological subject is good enough, exactly as good as any other behaviorally adequate description of it; you may prefer some over others on account of manipulative ease and computational cost, but the question of "reality" or "truth" does not arise. If this is your view of the nature of psychology, you will dismiss as meaningless the question which of the many behaviorally adequate psychologies is "really true" of the subject.

But if you adopt the perspective of "realism" on scientific theories, or at any rate about psychology, you will not think all behaviorally adequate descriptions are psychologically adequate. An adequate psychology for the realist must have "psychological reality": That is, the internal states it posits must be the real states of the organism with an active role as causal intermediaries between sensory inputs and behavior outputs, and this means that only a Turing machine that is a correct machine description of the organism is an acceptable psychological theory. The simplest and most elegant behavioral description may not be the one that correctly describes the inner processes that cause the subject's observable behavior; there is no a priori reason to suppose that our subject is put together according to the specifications of the simplest and most elegant theory (whatever your standards of simplicity and elegance might be).

Why should one want to go beyond the instrumentalist position and insist on psychological reality? There are two related reasons: (1) Psychological states, namely, the internal states of the psychological subject posited by a psychology, must be regarded as real, as we saw, if we expect the theory to generate explanations, especially causal explanations, of behavior. And this seems to be the attitude of working psychologists: It is their common, almost universal, practice to attribute to their subjects internal states, capacities, functions, and mechanisms (for example, information processing and storage, reasoning and inference, mental imagery, preference structures) and to refer to them in formulating what they regard as causal explanations of behavior. Further, (2) it seems natural to expect—this seems true of most psychologists and cognitive scientists—to find actual neural-biological mechanisms that underlie the psychological states, capacities, and functions posited by correct psychological theories. Research in the neural sciences, in particular cognitive neuroscience, have had impressive successes—and we expect this to continue—in identifying physiological mechanisms that implement psychological and cognitive capacities and functions. It is a reflection of our realistic stance toward psychological

theorizing that we generally expect, and sometimes insist on, physiological foundations for psychological theories. The requirement that the correct psychology of an organism be a machine description of it,[21] not merely a behaviorally adequate one, can be seen as an expression of a commitment to realism about psychological theory.

If the psychology of any organism can be represented as a Turing machine, it is natural to consider the possibility of using representability by a Turing machine to explicate, or define, what it is for something to have a psychology. As we saw, that precisely is what machine functionalism proposes: What it is for an organism, or system, to have a psychology—that is, what it is for an organism to have mentality—is for it to realize an appropriate Turing machine. It is not merely that anything with mentality has an appropriate machine description; machine functionalism makes the stronger claim that its having a machine description of an appropriate kind is *constitutive of* its mentality. This is a philosophical thesis about the nature of mentality: Mentality, or having a mind, consists in being a physical computer that realizes a Turing machine of appropriate complexity and powers. What makes us creatures with mentality, therefore, is the fact that we are Turing machines. Having a brain is important to mentality, but the importance of the brain lies exactly in its being a computing machine. It is our brain's computational powers, not its biological properties and functions, that constitute our mentality. In short, our brain is our mind because it is a computing machine, not because it is composed of the kind of protein-based biological stuff it is composed of.

## Machine Functionalism: Further Issues

Suppose that two systems, $S_1$ and $S_2$, are *in the same mental state* (at the same time or different times). What does this mean on the machine-functionalist conception of a mental kind? A mental kind, as you will remember, is supposed to be an internal state of a Turing machine (of an "appropriate kind"); so for $S_1$ and $S_2$ to be in the same state, there must be some Turing machine state $q$ such that $S_1$ is in $q$ and $S_2$ is also in $q$. But what does this mean?

$S_1$ and $S_2$ are both physical systems, and we know that they could be systems of very different sorts (recall multiple realizability). As physical systems, they have physical states (that is, they instantiate certain physical properties); to say that they are both in machine state $q$ at time $t$ is to say this: There are physical states $Q_1$ and $Q_2$ such that $Q_1$ realizes $q$ in $S_1$, and $Q_2$ realizes $q$ in $S_2$, and, at $t$, $S_1$ is in $Q_1$ and $S_2$ in $Q_2$. Multiple realizability tells us that $Q_1$ and $Q_2$ need not have much in common qua physical states; one could be a biological

state and the other an electronic one. What binds the two states together is only the fact that in their respective systems they implement the same internal machine state. That is to say, the two states play the same computational role in their respective systems.

But talk of "the same internal machine state $q$" makes sense only in relation to a specific machine table. That is to say, internal states of a Turing machine are identifiable only relative to a particular machine table: In terms of the layout of machine tables we used earlier, an internal state $q$ is wholly characterized by the vertical column of instructions appearing under it. But these instructions refer to other internal states, say, $q_i$, $q_j$, and $q_k$, and if you look up the instructions falling under these, you are likely to find references back to state $q$. So these states are inter-defined. What all this means is that *the sameness or difference of an internal state across different machine tables—that is, across different Turing machines—has no meaning*. It makes no sense to say of an internal state $q_i$ of one Turing machine and a state $q_k$ of another Turing machine that $q_i$ is, or is not, the same state as $q_k$; nor does it make sense to say of a physical state $Q_i$ of a physically realized Turing machine that it realizes, or does not realize, the same internal machine state $q$ as does a physical state $Q_k$ of another physical machine, *unless the two physical machines are realizations of the same Turing machine*.

Evidently, then, the machine-functionalist conception of mental kinds has the following consequence: For any two subjects to be in the same mental state, they must realize the same Turing machine. But if they realize the same Turing machine, their total psychology must be identical. That is, on machine functionalism, two subjects' total psychology must be identical if they are to share even a single psychological state—or even to give meaning to the talk of their being, or not being, in the same psychological state. This sounds absurd: It does not seem reasonable to require that for two persons to share a mental state—say, the belief that snow is white—the total set of psychological regularities governing their behavior must be exactly identical. Before we discuss this issue further, we must attend to another matter, and this is the problem of how the inputs and outputs of a system are to be specified.

Suppose that two systems, $S_1$ and $S_2$, realize the same Turing machine; that is, the same Turing machine gives a correct machine description for each. We know that realization is relative to a particular input-output specification; that is, we must know what is to count as input conditions and what is to count as behavior outputs before we can tell whether it realizes a given Turing machine. Let $V_1$ and $V_2$ be the input-output specifications for $S_1$ and $S_2$, respectively, relative to which they realize the same Turing machine. Since the same machine

table is involved, $V_1$ and $V_2$ must be isomorphic: The elements of $V_1$ can be correlated, one to one, with the elements of $V_2$ in a way that preserves their roles in the machine table.

But suppose that $S_1$ is a real psychological system, perhaps a human (call him Larry), whereas $S_2$ is a computer, an electromechanical device (call it MAX). So the inputs and outputs specified by $V_2$ are the usual inputs and outputs appropriate for a computing machine, perhaps strings of symbols entered on the keyboard or images scanned by a video camera as input and symbols or images displayed on the monitor or its printout as output. According to machine functionalism, Larry and MAX have the same psychology. But shouldn't this strike us as absurd? One might say: MAX is only a computer simulation of Larry's psychology, and in granting MAX the full psychological status that we grant Larry, machine functionalism is *conflating a psychological subject with a computer simulation of it*. No one will confuse the operation of a jet engine or the spread of rabies in wildlife with their computer simulations. It is difficult to believe that this distinction suddenly vanishes when we perform a computer simulation of the psychology of a person. (We will return to this question below in a section on computationalism and the Chinese room argument.)

One thing that obviously seems wrong about our computer, MAX, as a psychological system when we compare it with Larry is its inputs and outputs: Although its input-output specification is isomorphic to Larry's, it seems entirely inappropriate for psychology. It may not be easy to characterize the differences precisely, but we would not consider inputs and outputs consisting merely of strings of symbols, or electronic images, as appropriate for something with true mentality. Grinding out strings of symbols is not like the full-blown behavior that we see in Larry. For one thing, MAX's outputs have nothing to do with its survival or continued proper functioning, and its inputs do not have the function of providing MAX with information about its surroundings. As a result, MAX's outputs lack what may be called "teleological aptness" as a response to its inputs. All this makes it difficult to think of MAX's outputs as constituting real behavior or action, something that is necessary if we are to regard it as a genuine psychological system.

Qua realizations of a Turing machine, MAX and Larry are symmetrically related. If, however, we see here an asymmetry in point of mentality, it is clear that the nature of inputs and outputs is an important factor, and our considerations seem to show that for a system realizing a Turing machine to count as a psychological system, its input-output specification (relative to which it realizes the machine) must be *psychologically appropriate*. Exactly what this appropriateness consists in is an interesting and complex question that requires

further exploration. In any case, the machine functionalist must confront this question: Is it possible to give a characterization of this input-output appropriateness that is consistent with functionalism—in particular, without using mentalistic terms or concepts? Recall a similar point we discussed in connection with behaviorism: Not to beg the question, the behavior that the behaviorist is allowed to talk about in giving behavioristic definitions of mental concepts must be "physical behavior," not intentional action with an explicit or implicit mental component (such as reading the morning paper, being rude to a waiter, or going to a music concert). If your project is to get mentality out of behavior, your notion of behavior must not presuppose mentality.

The same consideration applies to the machine functionalist: Her project is to define mentality in terms of Turing machines and input-output relations. The additional tool she can make use of, something not available to the behaviorist, is the concept of a Turing machine with its "internal" states, but her input and output are subject to the same constraint—her input-output, like the behaviorist's, must be physical input-output. If this is right, it seems no easy task for the machine functionalist to distinguish, in a principled way, Larry's inputs-outputs from MAX's, and hence genuine psychological systems from their simulations. We pointed out earlier that Larry's outputs, given his inputs, seem *teleologically apt*, whereas MAX's do not. They have something to do with his proper functioning in his environment—coping with the ever-changing conditions of his surroundings and satisfying his needs and desires. But can this notion of teleology—purposiveness or goal-directedness—be explained in a psychologically neutral way, without begging the question? Perhaps some biological-evolutionary story could be attempted, but it remains an open question whether such a bioteleological program will succeed. These considerations give credence to the idea that in order to have genuine mentality, a system must be embedded in a natural environment (ideally including other systems like it), interacting and coping with it and behaving appropriately in response to the new, and changing, conditions it encounters.

Let us now return to the question of whether machine functionalism is committed to the consequence that two psychological subjects can share a psychological state only if they have an identical total psychology. As we saw, the implication follows from the fact that, on machine functionalism, being in the same psychological state is being in the same internal machine state and that the sameness, or difference, of machine states makes sense only in relation to the same Turing machine, and never across distinct Turing machines. What is perhaps worse, it also follows that it makes no sense to say that two psychological subjects are *not* in the same psychological state unless they have

an identical total psychology! But this conclusion must be slightly weakened in consideration of the fact that the input-output specifications of the two subjects realizing the same Turing machine may be different and that the individuation of psychologies may have to be made sensitive to input-output specifications (we return shortly to this point). So let us speak of "isomorphic" psychologies for psychologies that are instances of the same Turing machine *modulo* input-output specification. We then have the following result: On machine functionalism, for two psychological subjects to share even a single mental state, their total psychologies must be isomorphic to each other. Recall Putnam's complaint against the psychoneural identity theory: This theory makes it impossible for both humans and octopuses to be in the same pain state unless they share the same brain state, an unlikely possibility. But we now see that the table is turned against Putnam's machine functionalism: For an octopus and a human to be in the same pain state, they must share an isomorphic psychology—an unlikely possibility, to say the least! And for two humans to share a single mental state, they must have an identical total psychology (since the same input-output specification presumably must hold for all or most humans). No analogous consequence follows from the psychoneural identity theory; in this respect, therefore, machine functionalism seems to fare worse than the theory it hopes to replace. All this is a consequence of a fact mentioned earlier, namely, that on functionalism, the individuation of mental kinds is essentially *holistic*; that is, what makes a given mental kind the kind it is depends on its relationships to other mental kinds, where the identities of these other mental kinds depend similarly on their relationships to still other mental kinds, and so on.

Things are perhaps not as bleak for machine functionalism, however, as they might appear, for the following line of response seems available: For both humans and octopuses to be in pain, it is not necessary that *total* octopus psychology coincide with, or be isomorphic to, *total* human psychology. It is only necessary that there be *some* Turing machine that is a correct machine description of both and in which pain figures as an internal machine state; it does not matter if this shared Turing machine falls short of the maximally detailed Turing machines that describe them (these machines represent their "total psychologies"). So what is necessary is that humans and octopuses share a partial, or abbreviated, psychology that covers pains (and perhaps also related sensations). Whether "pain psychology" can be so readily isolated, or abstracted, from a total psychology is a question worth pondering, especially in the context of the functionalist conception of mentality, but there is another related issue that we should briefly consider.

Recall the point that all this talk of humans' and octopuses' realizing a Turing machine is relative to an input-output specification. Doesn't this mean, in view of our earlier discussion of a real psychological subject and a computer simulation of one, that the input and output conditions characteristic of humans when they are in pain must be appropriately similar, if not identical, to those characteristic of octopuses' pains, if both humans and octopuses can be said to be in pain? Consider the output side: Do octopuses wince and groan in reaction to pain? They perhaps can wince, but they surely cannot groan or scream and yell "Ouch!" How similar is octopuses' escape behavior, from the purely physical point of view, to the escape behavior of, say, middle-aged, middle-class American males? Is there an abstract enough *nonmental* description of pain behavior that is appropriate for humans and octopuses and all other pain-capable organisms and systems? If there is not, machine functionalism seems to succumb again to the same difficulty that the functionalist has charged against the brain-state theory: An octopus and a human cannot be in the same pain state. Again, the best bet for the functionalist seems to be to appeal to the "teleological appropriateness" of an octopus's and a person's escape behaviors—that is, the fact that the behaviors are biologically appropriate responses to the stimulus conditions in enhancing their chances of survival and their well-being in their respective environments.

There is a further "appropriateness" issue for Turing machines that we must raise at this point. You will remember our saying that for a machine functionalist, a system has mentality just in case it realizes an "appropriately complex" Turing machine. This proviso is necessary because there are all sorts of simple Turing machines (recall our sample machines) that clearly do not suffice to generate mentality. But how complex is complex enough? What is complexity anyway, and why does it matter? And what kind of complexity is "appropriate" for mentality? These are important but difficult questions, and machine functionalism, unsurprisingly, has not produced detailed general answers to them. What we have, though, is an intriguing proposal, from Alan Turing himself, of a test to determine whether a computing machine can "think." This is the celebrated "Turing test," and this is the right time to consider Turing's proposal.

## Can Machines Think? The Turing Test

Turing's innovative proposal is to bypass these general theoretical questions about appropriateness in favor of a concrete operational test that can evaluate the performance capabilities of computing machines vis-à-vis average hu-

mans who, as all sides would agree, are fully mental.[22] The idea is that if machines can do as well as humans on certain cognitive, intellectual tasks, then they must be judged no less psychological ("intelligent") than humans. What, then, are these tasks? Obviously, they must be those that, intuitively, require intelligence and mentality to perform. Turing describes a game, the "imitation game," to test for the presence of these capacities.

The imitation game is played as follows. There are three players: the interrogator, a man, and a woman, with the interrogator segregated from the other two in another room. The man and woman are known only as "X" and "Y" to the interrogator, whose object is to identify which is the man and which is the woman by asking questions via keyboard terminals and monitors. The man's object is to mislead the interrogator to make an erroneous identification, whereas the woman's job is to help the interrogator. There are no restrictions on the topics of the questions asked.

Suppose, Turing says, we now replace the man with a computing machine. The machine is programmed to simulate the part played by the man to fool the interrogator into making wrong guesses. Will the machine do as well as the man in fooling the interrogator? Turing's proposal is that if the machine does as well as the man, then we must credit it with all the intelligence that we would normally confer on a human; it must be judged to possess the full mentality that humans possess.[23]

The gist of Turing's idea can be captured in a simpler test: By asking questions (or just holding a conversation) via keyboard terminals, can we find out whether we are conversing with a human or a computing machine? (This is the way the Turing test is now being conducted.) If a computer can consistently fool us so that our success in ascertaining its identity is no better than what could be achieved by random guesses, we must concede, it seems, that this machine has the kind of mentality that we grant to humans. There already are chess-playing computers that would fool most people this way, but only in playing chess: Average chess players would not be able to tell if they are playing a human opponent or a computer. But the Turing test covers all possible areas of human concern: politics and sports, music and poetry, how to fix a leaking faucet or make a soufflé—no holds are barred.

The Turing test is designed to isolate the questions of intelligence and mentality from irrelevant considerations, such as the appearance of the machine (as Turing points out, it does not have to win beauty contests), details of its composition and structure, whether it speaks and moves about like a human, and so on. The test is to focus on a broad range of rational, intellectual capacities and functions. But how good is the test?

Some have objected that the test is too tough and too narrow. Too tough because something does not have to be smart enough to outwit a human to have mentality or intelligence; in particular, the possession of a language should not be a prerequisite for mentality (think of mute animals). Human intelligence itself encompasses a pretty broad range, and there appears to be no compelling reason to set the minimal threshold of mentality at the level of performance required by the Turing test. The test is perhaps also too narrow in that it seems at best to be a test for the presence of *humanlike* mentality, the kind of intelligence that characterizes humans. Why couldn't there be creatures, or machines, that are intelligent and have a psychology but would fail the Turing test, which, after all, is designed to test whether the computer can fool a *human* interrogator into thinking it is a *human*? Furthermore, it is difficult to see it as a test for the presence of mental states like sensations and perceptions, although it may be a good test of broadly intellectual and cognitive capacities (reasoning, memory, and so on). To see something as a full psychological system we must see it in a real-life context, we might argue; we must see it coping with its environment, receiving sensory information from its surroundings, and behaving appropriately in response to it.

Various replies can be attempted to counter these criticisms, but can we say that the Turing test at least provides us with a *sufficient* condition for mentality, although, for the reasons just stated, it cannot be considered a *necessary* condition? If something passes the test, it is at least as smart as we are, and since we have intelligence and mentality, it would be only fair to grant it the same status—or so we might argue. This reasoning seems to presuppose the following thesis:

> *Turing's Thesis*. If two systems are input-output equivalent, they have the same psychological status; in particular, one is mental, or intelligent, just in case the other is.

We call it Turing's Thesis because Turing appears to be committed to it. Why? Because the Turing test looks only at inputs and outputs: If two computers produce the same output for the same input, for all possible inputs—that is, if they are input-output equivalent—their performance on the Turing test will be exactly identical,[24] and one will be judged to have mentality if and only if the other is. This means that if two Turing machines are correct behavioral descriptions of some system (relative to the same input-output specification), they are psychological systems to the same degree. In this way the general philosophical stance implicit in Turing's Thesis is more behavioristic than

machine-functionalist. For machine functionalism is consistent with the denial of Turing's thesis: It says that input-output equivalence, or behavioral equivalence, is not sufficient to guarantee the same degree of mentality. What follows from machine functionalism is only that systems that realize the same Turing machine—that is, systems for which an identical Turing machine is a correct machine description—enjoy the same degree of mentality.

It appears, then, that Turing's Thesis is mistaken: Internal processing ought to make a difference to mentality. Imagine two machines, each of which does basic arithmetic operations for integers up to 100. Both give correct answers for any input of the form $n + m$, $n \times m$, $n–m$, and $n \div m$ for whole numbers $n$ and $m$ less than or equal to 100. But one of the machines calculates ("figures out") the answer by applying the usual algorithms we use for these operations, whereas the other has a file in which answers are stored for all possible problems of addition, multiplication, subtraction, and division for integers up to 100, and its computation consists in "looking up" the answer for any problem given to it. The second machine is really more like a filing system than a computing machine; it does nothing that we would normally describe as "calculation" or "computation." Neither machine is nearly complex enough to be considered for possible mentality; however, the example should convince us that we need to consider the structure of internal processing, as well as input-output correlations, in deciding whether a given system has mentality.[25] If this is correct, it shows the inadequacy of a purely behavioral test, such as the Turing test, as a criterion of mentality.

So Turing's Thesis seems incorrect: Input-output equivalence does not imply equal mentality. But this does not necessarily invalidate the Turing test, for it may well be that given the inherent richness and complexity of the imitation game, any computing machine that can consistently fool humans—in fact, any machine that is in the ballpark for the competition—has to be running a highly sophisticated, unquestionably "intelligent" program, and there is no real chance that this machine could be operating like a gigantic filing system with a superfast retrieval mechanism.[26] We should note that the computing machines' performance at actual Turing tests—and these have been restricted tests, on specific topics—has been truly dismal so far; computers programmed to fool human judges have not come anywhere near their goal. Turing's prediction in 1950 that in fifty years we would see computers passing his test has missed the mark—by a huge margin. It is also true, though, that designing a "thinking" machine that will pass the Turing test has not been a priority for artificial-intelligence researchers for the past several decades.

## Computationalism and the "Chinese Room"

Computationalism, or the computational theory of mind, is the view that cognition, human or otherwise, is information processing, and that information processing is computation over symbolic representations according to syntactic rules, rules that are sensitive only to the shapes of these representations. This view of mental, or cognitive, processes, which arguably is the reigning research paradigm in many areas of cognitive science, regards the mind as a digital computer that stores and manipulates symbol sequences according to fixed rules of transformation. On this view, mental events, states, and processes are computation events, states, and processes, and there is nothing more to a cognitive process than what is captured in a computer program successfully modeling it. This perspective on minds and mental processes seems to entail—at least, it encourages—the claim that a computer running a program that models a human cognitive process is itself engaged in that cognitive process. Thus, a computer that successfully simulates college students constructing proofs in sentential logic is itself engaged in the activity of constructing logical proofs. As we saw earlier, machine functionalism holds that having a mind is being a physical Turing machine of appropriate complexity. The issue of "appropriateness" aside, it is clear that the route from machine functionalism to computationalism is pretty straight and short.

This view of computation and mind is what John Searle calls "strong AI," which he characterizes as follows:

> According to strong AI, the computer is not merely a tool in the study of the mind; rather, the appropriately programmed computer really is a mind, in the sense that computers given the right programs can be literally said to *understand* and have other cognitive states.[27]

Before we discuss Searle's intriguing argument against computationalism, we might wonder why anyone would conflate a simulation with the real thing being simulated. Computers are used to simulate many different things: the performance of a jet engine, the spread of rabies in wildlife, the progress of a hurricane, and on and on. But no one would confuse a computer simulating a jet engine with a jet engine, or a computer simulation of a tornado with a tornado. So why would anyone want to say that a computer simulation of a cognitive process is itself a cognitive process? Isn't this a simple confusion? The following answer is open to the computationalist: It is because a cognitive process itself is a computational process. This means that a computer simula-

tion of a cognitive process is a simulation of a computational process, and obviously a computational simulation of a computational process is to re-create that computational process. Thus, there is no confusion in the claim that a computer simulating a cognitive process is itself engaged in that cognitive process. But this response makes sense only if we have already accepted the claim that cognitive processes are computational processes—that is, the truth of computationalism. This is the view of mind that Searle's Chinese room argument is expressly designed to refute.

To prepare us for his argument, Searle describes a program developed by Roger Schank and his colleagues to model our ability to understand stories. It is part of this ability that we are able to answer questions about details of a story that are not explicitly stated. Searle gives two examples. In the first story you are told: "A man goes into a restaurant and orders a hamburger. When it arrives, it is burned to a crisp, and the man angrily leaves, without paying for the hamburger." If you are asked "Did the man eat the hamburger?" you would presumably say "No." The second story goes like this: "A man goes into a restaurant and orders a hamburger. When it arrives, he is very pleased, and when he leaves, he leaves a big tip for the waiter." If you are asked "Did the man eat the hamburger?" you would say "Yes, he did." Schank's program is designed to answer questions like these in appropriate ways when it is given the stories. To do this, it has in its memory information concerning restaurants and how people behave in restaurants, ordering dishes, tipping, and so on. For the sake of the argument, we may assume Schank's program works flawlessly—it works perfectly as a simulation of the human ability to understand stories. The claim of computationalism would then be that a computer running Schank's program *literally understands* stories, just as we do.[28]

To undermine this claim, Searle constructs an ingenious thought-experiment.[29] He invites us to imagine a room—the "Chinese room"—in which someone (say, Searle himself) who understands no Chinese is confined. There are two piles of Chinese texts in the room, one called "the script" (this corresponds to the background information about restaurants, etc., in Schank's program) and "the story" (corresponding to the story on which understanding is tested). Searle is provided with a set of rules stated in English (the "rule book," which corresponds to Schank's program) for systematically transforming strings of symbols, by consulting the script and the story, to yield further symbol strings. These symbol strings are made up of Chinese characters, and the transformation rules are purely *formal*, or *syntactic*, in that their applications depend solely on the shapes of the symbols involved, not their meanings. So you can apply these rules without knowing any Chinese (remember: the rules

are stated in English); all that is required is that you have the ability to recognize Chinese characters by their shapes. Searle becomes very adept at manipulating Chinese expressions in accordance with the rules given to him (we may suppose that Searle has memorized the whole rule book) so that every time a string of Chinese characters is sent in, Searle goes to work, consulting the two piles of Chinese texts in the room, and promptly sends out a string of Chinese characters. From the perspective of someone outside the room who knows Chinese, the input strings are questions in Chinese about the story and the output strings sent out by Searle are responses to these questions. The input-output relationships are what we would expect if a Chinese speaker, instead of Searle, were locked inside the room. And yet Searle does not know any Chinese and does not understand the story, and there is no understanding of Chinese going on anywhere inside the room. What goes on is only manipulation of symbols on the basis of their shapes, or "syntax," but real understanding involves "semantics," knowing what these symbols represent, or mean. Although Searle's behavior is input-output equivalent to that of a speaker of Chinese, Searle knows no Chinese and does not understand the story (remember: the story is in Chinese).

Now, replace Searle with a computer running Searle's rule book as its program. This changes nothing: Both Searle and the computer are syntax-driven machines manipulating strings of symbols according to their shapes. In general, what goes on inside a computer is exactly like what goes on in the Chinese room (with Searle in it): rule-governed manipulations of symbols based on their syntactic shapes. There is no more understanding of the story in the computer than there is in the Chinese room. The conclusion to be drawn, Searle argues, is that mentality is more than rule-governed syntactic manipulation of symbols and that there is no way to generate semantics—what the symbols mean or represent—from their syntax. This means that understanding and other intelligent mental states and activities cannot arise from mere syntactic processes. Anyway, that is Searle's Chinese room argument.

Searle's argument has elicited a large number of critical responses, and just what the argument succeeds in showing remains controversial. Although its intuitive appeal and power cannot be denied, we have to be cautious in assessing its significance. The appeal of Searle's example may be due, some have argued, to certain misleading assumptions tacitly made in the way he describes what is going on in the Chinese room. Searle himself describes, and tries to respond to, six "replies" to his argument. Some of the objections to Searle raise serious points, and the reader is urged to examine them and Searle's responses. These responses are often ingenious and thought-provoking; however, Searle tends to

appeal to the intuitions of his readers, and we could do more, it seems, to drive home his central point, namely the thesis that syntactical manipulations do not generate meanings, or anything that can be called an understanding of stories. Consider, then, the following reconstructed argument in behalf of Searle:

> (1) Let us begin by asking what exactly is the difference between, on one hand, Searle/the computer in the Chinese room and, on the other, a Chinese speaker. (We assume that the program being run is Schank's program modeling story understanding.)
>
> (2) To understand the two stories in Chinese about a man ordering a hamburger in a restaurant, you must know, among other things, that "煎牛肉饼" means hamburger.
>
> (3) The Chinese speaker knows this, but neither Searle nor the computer does. That is why the Chinese speaker understands the stories, but Searle and the computer do not.
>
> (4) No amount of syntactic manipulation of Chinese characters will enable someone to acquire the knowledge of what "煎牛肉饼" means.
>
> (5) Hence, computationalism is false; neither Searle nor the computer running Schank's program understands the stories.

The central idea is that knowledge of meaning, or semantic knowledge, involves *word-to-world* (or *language-to-world*) relationships, whereas syntax concerns only properties and relations *within* a language as a symbol system. To acquire meanings, you must break out of the symbol system into the real world. Pushing symbols around according to their shapes will not get you in touch with extralinguistic reality. To know that "煎牛肉饼" means hamburger, you have to know what hamburgers are, and you come by this knowledge only through real-life contact with hamburgers (eating a few will help), or through descriptions in terms of other things you know through your real-life experience. Syntactic symbol manipulation alone will not yield such knowledge; only real-world experience will.

To expect syntactic operations to generate knowledge of meaning is like trying to learn a new language, say Russian, by memorizing a Russian–Russian dictionary. Or consider this example: You memorize a Korean-Japanese dictionary, and it may be possible for you to translate any Korean sentence into

Japanese by following a set of formal rules (stated in English—you can memorize these rules, too, like Searle memorizing the rule book). (Think of the translation programs available on many websites.) But you do not understand a word of Korean, or a word of Japanese, though you have become a proficient translator between the two languages. To understand either language, you have to know how that language is hooked up with the things in the real world.

So far so good. We have to be cautious, though, about what our argument, if successful, shows. It only shows that the computer running Schank's program (sitting in the basement of some computer-science lab) has no understanding of the stories in Chinese. It does not show, as Searle thinks the Chinese room shows, that no computing machine, an electromechanical device running programs, can acquire semantic knowledge of the sort displayed in (2) above. What our argument suggests is that for a computing machine (or anything else) to acquire this kind of knowledge, it must be placed in the real world, interacting with its environment, acquiring information about its surroundings, and possibly interacting with other agents like itself. In short, it must be an android, not necessarily humanlike in appearance, but an agent and cognizer in real-life situations, like Commander Data in the television series *Star Trek: The Next Generation*. (How meanings arise is itself a big question in philosophy of mind and language; see chapter 8 on mental content.)

Searle, however, is of the opinion that meaning and understanding can arise only in biological brains,[30] a position he calls "biological naturalism." On this approach, neural states, those that underlie thoughts, will carry representational contents. However, it seems clear that there are no relevant differences between neural processes and computational processes that could tilt the case in favor of biology over electronics. The fact is that the same neurobiological causal processes will go on no matter what these neural states represent about the world or whether they represent anything at all. That is, neural processes are no more responsive to meaning and representational content than are electronic computational processes. Local physical-biological conditions in the brain, not the distal states of affairs represented by neural states, are what drive neural processes. If so, isn't Searle in the same boat as Turing and other computationalists?

The question, therefore, is not what drives computational processes or neural processes. In neither do meanings or contents play a causal role; it is only the syntactic shapes of symbolic representations and the intrinsic physicochemical properties of the brain states that drive the processes. The important question is how these representations and neural states acquire meanings and intentionality in the first place. This is where the contact with the real world enters the picture: What we can conclude with some confidence

at this point is that such contact is crucial if a system, whether a human person or a machine, is to gain capacities for speech, understanding, and other cognitive functions and activities.

## For Further Reading

The classic source of machine functionalism is Hilary Putnam's "Psychological Predicates" (later reprinted as "The Nature of Mental States"). See also his "Robots: Machines or Artificially Created Life?" and "The Mental Life of Some Machines"; all three papers are reprinted in his *Mind, Language, and Reality: Philosophical Papers*, volume 2. The first of these is widely reprinted elsewhere, including *Philosophy of Mind: Classical and Contemporary Readings*, edited by David J. Chalmers, and *Philosophy of Mind: A Guide and Anthology*, edited by John Heil. Ned Block's "What Is Functionalism?" is a clear and concise introduction to functionalism. Putnam, the founder of functionalism, later renounced it; see his *Representation and Reality*, chapters 5 and 6.

For a teleological approach to functionalism, see William G. Lycan, *Consciousness*, chapter 4. For a general biological-evolutionary perspective on mentality, Ruth G. Millikan's *Language, Thought, and Other Biological Categories* is an important source.

For issues involving the Turing test and the Chinese room argument, see Alan M. Turing, "Computing Machinery and Intelligence"; John R. Searle, "Minds, Brains, and Programs"; and Ned Block, "The Mind as Software in the Brain." These articles are reprinted in Heil's *Philosophy of Mind*. Also recommended are Block, "Psychologism and Behaviorism," and Daniel C. Dennett, *Consciousness Explained*, chapter 14. Entries on "Turing Test" and "Chinese Room Argument" in the *Stanford Encyclopedia of Philosophy* are useful resources.

For criticisms of machine functionalism (and functionalism in general), see Ned Block, "Troubles with Functionalism," and John R. Searle, *The Rediscovery of the Mind*.

## Notes

1. Later given a new title "The Nature of Mental States."

2. Donald Davidson's argument for mental anomalism (chapter 7) also played a part in the decline of reductionism. See Davidson's "Mental Events."

3. At least some of them, for it could be argued that certain psychological states can be had only by materially embodied subjects—for example, feelings of hunger and thirst, bodily sensations like pain and itch, and sexual desire.

4. Unless, that is, the very idea of an immaterial mental being turns out to be incoherent.

5. The terms "realize," "realization," and "realizer" are explained explicitly in a later section. In the meantime, you will not go far astray if you read "P realizes M" as "P is a neural substrate, or base, of M."

6. This principle entails mind-body supervenience, which we characterized as minimal physicalism in chapter 1. Further, it arguably entails the thesis of ontological physicalism, as stated in that chapter.

7. See Ronald Melzack, *The Puzzle of Pain*, pp. 15–16.

8. Some have argued that this function-versus-mechanism dichotomy is pervasive at all levels, not restricted to the mental-physical case; see, for example, William G. Lycan, *Consciousness*.

9. As I take it, something like this is the point of Karl Lashley's principle of "equipotentiality"; see his *Brain Mechanisms and Intelligence*, p. 25.

10. To borrow Ned Block's question in "What Is Functionalism?" pp. 178–179.

11. As we shall see in connection with machine functionalism, there is another sense of "function," the mathematical sense, involved in "functionalism."

12. Strictly speaking, it is more accurate to say that having *the capacity to sense pain* is being equipped with a tissue-damage detector, and that pain, as an occurrence, is the activation of such a detector.

13. The distinction between "real" and "nominal" essence goes back to John Locke. A full explanation of these notions cannot be provided here. See Locke, *An Essay on Human Understanding*, Book III, chapters iii and vi. For helpful discussion see Nicholas Jolley, *Locke: His Philosophical Thought*, chapters 4 and 8.

14. When do two mousetraps count as instances of the same realizer and when do they count as instances of different realizers? What about pains and their realizers? These are significant questions. For helpful discussion see Lawrence Shapiro, *The Mind Incarnate*.

15. See, for example, B. F. Skinner, "Selections from *Science and Human Behavior*."

16. Machine functionalism in terms of Turing machines developed in sections below can deal with this problem as well; however, the Ramsey-Lewis method presented in chapter 6 is more intuitive and perspicacious.

17. A treatment of the mathematical theory of computability in terms of Turing machines can be found in Martin Davis, *Computability and Unsolvability*, and in George S. Boolos, John P. Burgess, and Richard C. Jeffrey, *Computability and Logic*.

18. Strictly speaking, this was a proposal, called the Church-Turing Thesis, rather than a discovery. It turned out that various proposed notions of "effective"

or "mechanical" calculability, including computability by a Turing machine, turned out to be mathematically equivalent, defining the same class of functions. The thesis was the proposal that these notions of effective calculable functions be taken as equivalent ways of defining "computable" functions. For details see the entry "Church-Turing Thesis" in the *Stanford Encyclopedia of Philosophy*.

19. See, for example, Hilary Putnam, "Psychological Predicates."

20. For a statement and defense of a position of this kind, see Bas Van Fraassen, *The Scientific Image*.

21. Is there, for any given psychological subject, a *unique* Turing machine that is a machine description (relative to a specification of input and output conditions), or can there be (perhaps there always must be) multiple, nontrivially different machine descriptions? Does realism about psychology require that there be a unique one?

22. Alan M. Turing, "Computing Machinery and Intelligence."

23. It is probably more reasonable to restrict the claim to cognitive mentality, leaving out things like sensations and emotions.

24. To do well on a real-life Turing test, the computers will need to have a real-time processing speed, in addition to delivering the "right" output (answers) for the given input (questions).

25. For an elaboration and discussion of this point, see Ned Block, "Psychologism and Behaviorism."

26. Daniel C. Dennett, *Consciousness Explained*, pp. 435–440.

27. John R. Searle, "Minds, Brains, and Programs," p. 235 in *Philosophy of Mind: A Guide and Anthology*, ed. John Heil.

28. Here we are setting aside an important question discussed earlier, namely that of psychological reality. Is Schank's program merely input-output equivalent to human understanding of stories, or does it in some relevant sense mirror the actual cognitive processes involved in human story understanding?

29. John R. Searle, "Minds, Brains, and Programs."

30. Or, says Searle, structures (even computers) that have the same causal powers as brains. My brain, in virtue of its weight, has the causal power of breaking eggs when dropped on them. But surely having this causal power cannot be relevant to mentality. So just what causal powers of a brain must a thing have in order to enjoy a mental life? Obviously, the brain's powers to generate and sustain a mental life! As it stands, therefore, Searle's apparent concession on the biological basis of mentality isn't very helpful.