Programs in the Explanation of Behavior
Author(s): Robert Cummins
Reviewed work(s):
Source: *Philosophy of Science*, Vol. 44, No. 2 (Jun., 1977), pp. 269-287
Published by: The University of Chicago Press on behalf of the Philosophy of Science Association
Stable URL: http://www.jstor.org/stable/187351
Accessed: 29/08/2012 16:11

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at
http://www.jstor.org/page/info/about/policies/terms.jsp

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

The University of Chicago Press and *Philosophy of Science Association* are collaborating with JSTOR to digitize, preserve and extend access to *Philosophy of Science.*

http://www.jstor.org

PROGRAMS IN THE EXPLANATION OF BEHAVIOR*

ROBERT CUMMINS

University of Michigan

The purpose of this paper is to set forth a sense in which programs can and do explain behavior, and to distinguish from this a number of senses in which they do not. Once we are tolerably clear concerning the sort of explanatory strategy being employed, two rather interesting facts emerge; (1) though it is true that programs are "internally represented," this fact has no explanatory interest beyond the mere fact that the program is executed; (2) programs which are couched in information processing terms may have an explanatory interest for a given range of behavior which is independent of physiological explanations of the same range of behavior.

The idea that programs of instructions explain behavior is now a commonplace in psychology. The literature which has grown up in information processing psychology since the early fifties is large and increasingly influential, so I need not document my claim that many psychologists think behavior is explained by programs. But, although the idea is now familiar, it is controversial; controversial, moreover, in a way which is bound to attract and deserve philosophical attention. For while there are many disagreements concerning *which* programs explain behavior, there is a much more fundamental disagreement concerning what *sort* of explanation is being offered, and even concerning whether what is being offered is or could be an explanation at all. A program, after all, is not a law; it is more like a narrative. And a play-by-play account of behavior, while useful data, can hardly be explanatory.

The purpose of this paper is to set forth a sense in which programs can and do explain behavior, and to distinguish this from a number of senses in which they do not. Once we are tolerably clear concerning the sort of explanatory strategy being employed, two rather interesting facts emerge: (1) though it is true that programs are "internally represented," this fact has no explanatory interest; (2) programs which are couched purely in information processing terms may have an explanatory interest for a given range of behavior which is independent of physiological explanations of the same range of behavior.

*Received September 1975; revised July, 1976.

Philosophy of Science, 44 (1977) pp. 269-287.

Copyright © 1977 by the Philosophy of Science Association.

1. Introduction. What might be meant, then, by the claim that programs explain behavior, or are theories of behavior?

The first point I want to make is that we are all quite familiar in everyday life with appeals to programs in explaining behavior. Suppose we want to know how little Johnny, a boy innocent of electronics, is able to build sophisticated audio equipment. We all know the answer: he follows the instructions in the manuals that come with the kits. He can do what each instruction specifies—i.e., he already has the capacities required by each instruction—and he can follow the list, in the sense in which this means simply adhering to the specified order. Anyone who does what the instructions specify in the order specified, whether knowingly or by sheer accident, winds up having completed a very sophisticated task.

What the manual does is analyze a certain sophisticated performance into unsophisticated performances in a sophisticated order. This allows Johnny to build an amplifier, but it also allows us to *explain how* Johnny is able to do such a thing given his meagre capacities. Any set of instructions—recipes, a teacher's rules for doing long division, the directions on your hot water heater for relighting the pilot light—can, with a slight change from the usual point of view, be seen as an explanatory analysis of a complex capacity. The source of explanatory power in these cases is obvious: ability to execute a sophisticated performance is reduced without remainder to abilities which are, relatively speaking, simple and antecedently understood.

So the appeal to programs in explaining behavior is a commonplace. And a little reflection on this commonplace has put us in a position to sharpen the claim we are investigating: programs explain behavioral *capacities,* and they do this by analyzing the exercise of a complex capacity into the organized exercise of relatively less problematic capacities. The question answered is, "How is the individual able to do such and such?", and in the process we are provided with a more or less detailed analytical description of what the individual does (can do) when he does such and such. The performance is sliced up into independently significant "steps" in a way that is evidently nonarbitrary in at least the minimal sense that not any old way of slicing will work: we may slice our original capacity up into capacities our individual has not got, or into capacities that are more problematic than the original.

With this much under our belts, it is only a short step to the recognition that appealing to a program in explanation of a behavioral capacity is an instance of one of the two standardly available strategies for explaining capacities of any kind. Let us detour briefly into general philosophy of science and give a rough description of these two strategies. An understanding of how they differ and how they are supposed to fit together will prove useful in the subsequent discussion.

2. The Explanation of Capacities.[1] Psychological capacities are a species of disposition. Familiar nonpsychological examples are bouyancy, solubility in water, and flexibility. To attribute a disposition to something is (in part) to say what it would do were certain conditions to hold: it would float or dissolve were it placed in water; it would bend were it stressed. The point of the subjunctive construction is that a thing may have a certain disposition even though it never satisfies the requisite conditions, and hence never *manifests* its disposition. Thus to attribute a disposition to something is to say that its behavior is subject to a certain law, a law special to that kind of thing. The law of a water soluble thing is: were it placed in water it would dissolve, *ceteris paribus*. This is not a universal law, and hence the question arises as to why the things that are subject to it *are* subject to it. There must be (and are) certain features peculiar to water soluble things which explain why *they are,* and other things *are not,* subject to this law. To explain a disposition, then, is to explain why the associated law holds of the disposed objects and not other things.[2]

There are two distinct strategies one may employ in explaining a capacity. The first of these I call the *Subsumption Strategy*. This strategy should be familiar from chemistry and physics, and a single illustration should make it sufficiently clear what I have in mind. Consider the simple disposition that Brian O'Shaughnessy ([7]) calls elevancy: the tendency of an object to rise in water of its own accord. (Elevancy, of course, is not the same as bouyancy; concrete sailboats are bouyant, but not elevant.) To explain elevancy, we must explain why freeing a submerged elevant object causes it to rise. This we may do as follows. In every case, the ratio of an elevant object's mass to its nonpermeable volume is less than the mass per unit volume

[1] This matter is discussed from a different perspective in [4].

[2] The dispositions I have in mind here are ordinary household dispositions such as flexibility, together with the standard dispositions treated in chemistry and physics texts, e.g., acidity conceived as the capacity to "donate" protons. I wish to rule out such things as the tendency of masses to coalesce in space according to an inverse square law (which may not be a disposition at all (see [2])), and philosopher's inventions such as the "disposition" a certain bit of beach has to be covered by my body on a certain day (given "appropriate conditions," e.g., sunshine on the crucial day and my intention to go to the beach on that day if it is fine). Concentration on cases like these is unlikely to be helpful (unless our question is, "Why, exactly, *aren't* these genuine full-fledged dispositions?"). For more on the explanation of dispositions, see [1].

of water. Archimedes' Principle tells us that water exerts on a submerged object an upward force equal to the weight of the water displaced. In the case of an elevant object, this force evidently exceeds the weight of the object by some amount f. Freeing the object changes the net force on it from zero to a net force of magnitude f in the direction of the surface, and the object rises accordingly. Here we subsume the connection between freeings and risings under a general law connecting changes in net force with changes in motion, and we do this by citing a feature of elevant objects that allows us (*via* Archimedes' Principle) to exhibit freeing them under water as an instance of introducing a net force in the direction of the surface.[3]

The Subsumption Strategy is evidently of little use in explaining psychological capacities. Perhaps certain reflexes can be handled in this way, but little else can be expected to yield to this strategy without further ado.

The further ado in question is what I call the *Analytical Strategy*. Rather than subsume the dispositional regularity under a law not special to the disposed objects, the Analytical Strategy proceeds by analyzing a disposition into a number of other relatively less problematic dispositions such that organized manifestation of these analyzing dispositions amounts to a manifestation of the analyzed disposition. Schematic diagrams in electronics provide a familiar and transparent example of this sort of analysis in a physical science context. Since each symbol represents any physical object whatever having a certain capacity, a schematic diagram of a complex device constitutes an analysis of the electronic capacities of the device as a whole into the capacities of its components. Such an analysis allows us to explain how the device as a whole exercises the analyzed capacity, for it allows us to see exercises of the analyzed capacity as programmed exercises of the analyzing capacities. In this case, the "program" is given by the lines indicating how the components are connected, together with such statements as Ohm's law.

[3] I have called this strategy the Subsumption Strategy because I need a name for it, and 'subsumption' captures one central element in the strategy which is absent from, and irrelevant to, the other strategy I want to discuss. But the name is potentially misleading in that it might suggest that mere subsumption under law is all that is involved. That more than mere subsumption is involved is brought out as follows. We can easily imagine a property Φ and a disposition D such that 'All and only the things having D have Φ' is both true and lawlike, yet such that the presence of Φ does not explain the regularity associated with D. For instance, 'All and only the acids turn litmus red' is true and lawlike, yet we cannot explain the disposition to "donate" protons by appeal to this law. I have treated the contrast between explaining a disposition and merely subsuming it in somewhat more detail, though in a different context, in [3].

Functional analysis in biology is essentially similar. The biologically significant capacities of an entire organism are explained by analyzing the organism into a number of "systems"—the circulatory system, the nervous system, etc.—each of which has its characteristic capacities. These capacities are in turn analyzed into capacities of component organs and structures. Ideally, this strategy is pressed until pure physiology takes over, i.e., until the analyzing capacities are amenable to the Subsumption Strategy. We can easily imagine biologists expressing their analyses in a form analogous to the schematic diagrams of electrical engineering, with special symbols for pumps, filters, pipes, and so on.[4]

A natural assumption—and a correct one I think—is that the Analytical Strategy must eventually terminate in dispositions which yield to the Subsumption Strategy. For without this assumption, the apparent explanatory progress afforded by the Analytical Strategy is *mere* appearance. That strategy makes progress only insofar as the analyzing capacities are relatively less problematic as compared to the capacity analyzed. We undermine such progress if we suppose that our analyzing capacities might ultimately prove resistant to the Subsumption Strategy, for to suppose this is to allow that these capacities may be utterly mysterious and inexplicable from the point of view of physical science: we shall be barred from any account of why some things, and not others obey the associated law. One needn't endorse any starry-eyed claims about the unity of science to find this prospect unwelcome.[5]

Ultimate applicability of the Subsumption Strategy thus constitutes a constraint on particular applications of the Analytical Strategy. This is of some importance, for we shall see shortly that it is difficult to make any clear *sense* of this constraint, let alone satisfy it, when an information processing program is appealed to in explaining a psychological capacity. I think a more or less vague sense of this problem underlies much of the skepticism such appeals have aroused.

[4] For a more detailed discussion of the Analytical Strategy in a somewhat different context, see [4].

[5] The prospect is especially unwelcome in psychology for the following reason. Most capacities of interest to psychologists are or can be acquired (or lost) in ways more or less familiar to learning theorists, and this applies to capacities which are primitive from the point of view of analysis as well as to capacities which are analytically complex. Now it seems clear that the *acquisition* (or loss) of an analytically primitive capacity will be inexplicable unless we can see it in terms of a relatively permanent physical change in the organism: the onset of the capacity must stand to some physical change as, for instance, the onset of elevancy in the football stands to its inflation. (For a brief treatment of this point, see [1].)

3. Following Instructions: Two Senses. So much for our detour into general philosophy of science. We may return now to the specific problem at hand, namely clarifying the claim that programs explain behavior. Let us review the plot. First, we have found that it is a commonplace to explain a behavioral capacity by appeal to a program. Second, we have seen that this is an instance of a familiar and respectable explanatory strategy. And finally, we have achieved some grip on what this strategy is supposed to accomplish and how it is supposed to accomplish it. We are now prepared for more heady matters.

It is useful to recognize that recipes, manuals of instructions, rules for doing long division and the like can and do play an explanatory role in addition to the heuristic roles they are designed to play. It is useful if only because it shows that appealing to a program in explaining a behavioral capacity is not a novel and mysterious strategy invented by computer zealots. On the other hand, such examples can be extremely misleading in that the recipe or manual is an object in the environment: a complex stimulus which directs behavior in addition to merely analyzing behavior and describing its direction. Nothing is more obvious, however, than that an organism does not exercise its psychologically interesting capacities by consciously following an external program of instructions. This obvious fact has led some, presumably to preserve the analogy intact, to suppose that the required program is ''there'' all right, but ''internally represented'' in the brain, and ''tacitly known and followed.'' Others, rightly suspicious of this talk of ''internalization,'' have been led to reject the appeal to programs altogether.

From our present vantage point, this dispute should seem odd. We have just seen that appealing to a program to explain a capacity is an instance of an explanatory strategy which makes no use of the notion of being instructed to do something. We could express our analyses of electronic circuits explicitly in program form rather than in schematic diagrams, but it would evidently be pointless to suggest that the circuit does what it does because an ''internally respresented'' program directs its performance, telling it what to do when. This suggests that theorists who speak thus of internalization may have lost their grip on the point of their own strategy.

We can begin to sort this out if we distinguish two senses in which something or someone might be said to ''follow instructions.'' The more familiar sense, though by no means the better understood sense, is what we might call the *imperative sense:* the individual does what the instructions say to do, and does these things *because he is so instructed.* Thus recipes, manuals, etc. The other sense we might call the *descriptive sense:* the individual or system does what the instructions say to do in the order specified and so on, but not necessarily because he or it is so instructed. It is the possibility of taking programs as descriptive in this way which allows us to speak sensibly of brains and machines executing programs, for this sense does not have the implication that the individual or whatever does what it does because of the program. Unlike the recipe case, the program is an analytical description of what happens only. It is a theorist's tool, not a cause. This point is regrettably obscured—perhaps fudged—by the use of such phrases as "internalizing a program." Such phrases encourage—and often incorporate—a confusion between an analysis of a capacity and a cause of its exercise. The inevitable result is complete mystification as to what sort of explanation is being offered. Conversely, once we are clear about the explanatory strategy being employed, we find we have no need of, and no place for, the idea that the program actually directs performance in addition to describing and analyzing its direction.

In at least one clear sense, then, to claim that a thing—an organism or brain or whatever—can execute a program is to claim that exercise of a certain relatively sophisticated capacity of that thing can be analyzed into the organized exercise of certain relatively less sophisticated capacities of that thing or its parts. Since the program specifies the organization in question, it evidently provides a kind of narrative description of performance as well. It describes what happens in what order, and identifies these happenings as exercises of certain relatively simple capacities. But it does not explain *why* these capacities are exercised in the order specified, whereas the fact that the cook is looking at a printed recipe does help explain just this sort of fact.

Applications of the Analytical Strategy become more interesting as the gap in sophistication and type between analyzing and analyzed capacities grows large. When the gap is large, we trade sophisticated abilities for sophisticated organization. This is the idea behind the assembly line, and it is the idea behind Watson's treatment of habit as a sequence of conditioned reflexes([8]). As organization becomes more sophisticated, pressure grows to explain why things happen in the order they do. The pressure isn't very great in Watson's case because each response is supposed to produce the next stimulus in a physiologically unproblematic way. But it is important to see that *some* such supposition is required. As the organization grows in complexity, the question becomes correspondingly more difficult to answer. But it is a perfectly good question, and the need for an answer is only obscured by modes of speech which suggest that an internalized program directs matters. To say a capacity is explained by the fact that a certain program is "internalized" suggests that, having discovered *which* program is executed, we needn't explain *why* it is executed, i.e., why matters follow the particular course specified. This is trying to have the recipe and eat it too. Something causes events to take the course specified by the program, but the program itself is not the responsible party.

Now it might seem that this must be wrong: surely a computer does some of the things it does because it is programmed to do them, and surely to program a computer is to bring it about that a certain program is internally represented in the machine. Or again, surely it is sometimes right to say that a cook does some of the things he does because of a certain recipe; the recipe *directs* his behavior—indeed I said this myself a few pages back. Now when the cook has memorized the recipe, surely it cannot be wholly wrong to say that his *memory* directs his behavior, that he does what he does because of his memory, and isn't his memory of the recipe an internal representation of the recipe?

I think there is a way of interpreting these remarks so as to preserve their truth, but no way of interpreting them which preserves their interest as well. Let us begin by asking what is required for a program to be represented in a device or system. I think this matter is a good deal less complicated than has often been supposed, for it seems that for a program *P* to be represented in a system *S* it is sufficient that *S executes P.*[6] To see that it is sufficient, notice that there must be some relatively permanent structural features of *S* which ultimately account, *via* the Subsumption Strategy, for *S*'s capacity to execute *P*, structural features the acquisition (or loss) of which accounts for the acquisition (or loss) of that capacity. Now it seems that it must always be possible to harness these features as the required representations, each instruction being assigned to whatever structural features of *S* account (*via* the Subsumption Strategy) for the capacity that instruction specifies.[7] Once this assignment is made, we are free to think of the structural features underlying a given instruction as encoding that instruction, i.e., as an alternate symbolic representation of it.

Now let us imagine a certain system *S* which executes a certain program *P*, and hence represents *P* as well. In the execution of *P*

[6] Perhaps I should say explicitly here that, as I use the term, to say *S executes P* is to attribute a capacity to *S*. 'executes' is therefore to be distinguished from 'is executing', 'has executed', etc.

[7] If *S* is capable of following the program, then it is capable of following each instruction. Hence, each instruction expresses a capacity of *S* (or one of its parts).

by *S*, events take the course they do because *S* is structured in a certain way. Evidently *S* is not structured in this way *because P is represented in S*. Hence, if we are to insist that events take the course they do because *P* is represented in *S*, we must say that *S*'s being structured in the way it is just *is P*'s being represented in *S*. This seems plausible enough. Of course, it will not always be the case that a system representing a certain program executes that program, a written token of the program itself being an example of a system which represents the program but does not execute it. And even a system which does execute the program may represent that program in virtue of features which are quite irrelevant to execution, e.g., in virtue of the instructions being printed on various parts. But this problem may be avoided in a natural way by restricting attention to a stronger concept of representation which requires that each instruction be represented by those structural features which account for the system's capacity to execute it. We have just seen that systems that *do* execute the program are bound to represent it in virtue of the very facts which explain execution, and hence these systems are bound to strongly represent the program. Taking advantage of this, we can see that to say that *P* is (strongly) represented in *S* could be thought of as a way of attributing to *S* the structure which accounts for execution of *P* by *S*. Construed along these lines it is true that in the execution of *P* by *S* events take the course they do because *P* is (strongly) represented in *S*.

It is true . . ., but not interesting. To say that *S* strongly represents *P* adds nothing to the fact that *S* executes *P* beyond what is already required by our methodological constraint, viz., that, like all dispositions, execution ultimately be explicable (*via* the Subsumption Strategy) by appeal to structural features of *S*. No constraint is put on what these features must be beyond what is already imposed by the requirement that *S*'s capacity to follow *P* be explained. Indeed, we cannot say whether a representation *R* of *P* in *S* is a strong representation until we have determined whether, for each instruction, the features of *S* which *R* assigns to that instruction are such as to account for *S*'s capacity to execute that instruction.

If this is correct, then, though there is a sense in which it is true to say that a system *S* is capable of doing what a certain program *P* specifies because *P* is represented in *S*, saying this cannot have any explanatory force whatever. We do explain a complex capacity in a perfectly familiar way when we exhibit performance as execution of a program. But we cannot go on to explain why the program is executed, i.e., why events take the direction specified, by appealing to the fact that the program is represented in the executing system.[8]

Once persuaded of this result we must ask why the stories about programming a computer and memorizing a recipe seemed so compelling.[9] What we were able to salvage from these stories is this: for a computer to be programmed or a recipe memorized is just for the program or recipe to be represented in the computer or cook by the very structural features which account for the capacities in question. This evidently leads nowhere, yet the examples are surely not entirely without point. For one thing, when we program a computer we do something to it—feed in cards or whatever—which *alters* it in a relatively permanent way, i.e., in a way which endures until the machine is programmed again, something wears out or the like. Now it is perfectly in order to say that the computer executes the program because it was altered in a certain way, and this seems to be part of what underlies the misleading remark that the computer does what it does—i.e., follow the program—because it is programmed to do it. And perhaps something analogous partly underlies the equally misleading remark that the cook follows the recipe because he has memorized it.

Another thing that is at work here is an adumbrated but genuine application of the Analytical Strategy. When we say that the cook stirs the candy while heating it because the recipe he has memorized required this, *part* of what we do is point to a recipe, or the existence of a recipe, which exhibits the role or function of stirring while heating in relation to the larger project the cook is engaged in when he stirs while heating. In the same way, a flow chart for doing long division tells us what bringing down the next digit contributes to getting a quotient. If someone who follows (descriptively) this chart when he finds quotients now "brings down the next digit," we may explain why he does what he does in the sense in which this means explaining what his current doing contributes to his larger undertaking. This sort of fact is not unnaturally expressed by saying that he is bringing down the next digit because the chart requires it at this point. But its naturalness does not prevent it from being dangerous in the hands of philosophers and psychologists.

[8] For a more detailed treatment of this point, see the Appendix.

[9] To these stories we might add the following. "For a program *P* to be represented in *S* is (sometimes) for *S* to have certain *information* available. And certainly it can't be wholly wrong to say that *S* can do certain things because certain information is available to *S*." Of course this is right, provided the "things *S* can do because the information is available" are not the very things specified by *P*, for, otherwise to say the information is available will just be a misleading way of saying that *S* is structured in a way which explains its capacity to follow *P*.

So there is something—perhaps two things—legitimate in talk of computers doing what they do because they are programmed to do it, or cooks doing what they do because they have memorized a recipe which requires it. But it is not what the words lead us to expect when we take them as the sort of context-free expressions of literal truth beloved of scientists and philosphers. When we get a glimmer of what *is* legitimate in such talk we find that it lends no support whatever to the idea that execution is explained by representation.

4. Information Processing Programs.

(A) *Skeptical preliminaries.* By now it should seem more or less obvious that organisms—and indeed devices of all kinds—execute programs, and that complex capacities can be explained in a certain familiar and respectable sense by appeal to the fact that certain programs are executed. Indeed, once we see that this claim does not involve trying to have our recipe and eat it too, it seems that we are left with something no one could possibly deny. This is quite right, I think, provided we neglect the kind of programs everyone finds most interesting, namely information processing programs (hereafter often abbreviated IP)—programs for manipulating symbols.

Executing a program in the descriptive sense we have staked out seems a simple matter: a device—be it brain or computer—executes the program if it does what each instruction says to do and it does these things in the order specified.[10] So program execution seems to come down to instruction execution, and to execute an instruction in the descriptive sense is just to do what the instruction says to do.

The problem with this thought is that, depending on what sort of instruction is under consideration, it may be far from clear what counts as doing what that instruction says to do. It is perhaps clear enough whether some device is, say, closing the relays labeled '*A*' through '*D*,' and hence clear enough whether it is executing the instruction, 'CLOSE RELAYS *A* THROUGH *D*.' But how about, 'CARRY ALL BUT THE LEAST SIGNIFICANT DIGIT'? If our

[10] A more precise formulation would run as follows: were d to execute instructions 1 through n, then normally d would subsequently execute instruction $n + 1$, for all $0 < n < m$, where there are m instructions. Even this is faulty: what counts as the "next instruction" depends on which instructions have already been executed, and on the initial state of the device. The initial state determines the first instruction, and together these determine the second instruction, and so on. In this way, different initial states determine different *paths* through the program. The definition above, then, may be taken to define path execution, execution then being defined thus: d executes P iff d executes each path through P.

program consists of instructions like this, what sense can we make of the claim that a bunch of relays or flip-flops or neurons does what the program says to do? Perhaps we can explain, *via* the Subsumption Strategy, the electro-chemical capacities of neurons, but this will not help us to satisfy the constraint we placed on analysis if the capacities our analysis appeals to are capacities to manipulate symbols. Come to think of it, do we even know what it would be *like* to explain a capacity of that kind *via* the Subsumption Strategy? It is evident that there is no set of physical features a thing must have to have a capacity to perform a given symbolic operation: all adding machines add, but they are physically as disparate as wind-up alarm clocks and transistor radios. Indeed, it is hard to see how the physical facts could bear at all on whether or not a device executes a program of information processes, for such programs say nothing whatever about physical make up. Early researchers were quite clear about this matter. Thus Newell, Shaw and Simon reporting on the "Logic Theorist" in [6], stressed that the theory they were proposing was entirely neutral with respect to the physical properties a thing must have for their theory to be true of it. This seems to flout our constraint. The problem raised here has already been hinted at: no matter how elementary a symbolic transaction is, specifying a capacity to perform it is not specifying a physical disposition, and this makes it difficult to see how the Subsumption Strategy could ever get a grip on the atomic capacities such a program deals with.

(B) *Salvage.* Of course I have been willfully dense in the foregoing in order to bring out a certain problem. I have been exhibiting what one of my graduate students once called a prejudice in favor of thin symbols. If a smear of ink can be a numeral, i.e., represent a number, why not a closed relay or a neural connection?

This is fair enough, and useful up to a point. Evidently, in the right circumstances, an execution of the instruction to close relays *A* through *D* could count as execution of the instruction to carry all but the least significant digit. But under what circumstances? Well, very roughly and intuitively, closing relays *A* through *D* must stand to other transactions in the device as the instruction to carry all but the least significant digit stands to the other instructions in the program. There must be, in some sense yet to be explained, an "isomorphism of structure" between the information processing program and some program couched in physical terms. Even this very crude formulation is enough to make it clear that the concept of execution for IP programs is a tricky affair. The problem is that, given any physical transaction you like, and any symbolic operation you like, there will generally be some set of conditions—some context—in which that physical transaction would count as a performance of that symbolic operation. This follows more or less obviously from the reflection that the "fat symbols"—neurons or whatever—are, from our point of view, in code. In a cypher, any numeral can, taken independently, be assigned any significance whatever. It is only a definite context which places any constraint on the significance to be assigned to an individual numeral, the requirement being that when each numeral is assigned a meaning by a determinate rule, a coherent message should result. We cannot get at program execution one instruction at a time for reasons exactly analogous to those preventing us from getting at cypher significance one numeral at a time.

Actually, cyphers are easier than nervous systems in two respects. First, although there are indefinitely many different ways to make sense of a cypher if there are any ways at all, what we are after is the intended message. Thus there is a unique right answer among the infinity of workable solutions, and we generally can tell, given the context, whether a given solution is the right one by this criterion. But when we are attempting to treat transactions in the nervous system as symbolic operations, there is no "intended interpretation": we are seeking an interpretation which will be theoretically fruitful, and, as is the case with scientific description generally, there is no way to tell in advance whether a given way of describing matters will prove a help or a hindrance. The fact that workable solutions are not unique, and that the first one we hit on may not be the best, is often forgotten simply because it is so difficult to come up with any workable solution at all. But there is no reason to suppose that the criterian of theoretical fruitfulness selects a unique solution as *the correct solution.*

The second respect in which cyphers are easier is more serious: we know how to individuate numerals in standard notation, and we know that numerals are the significant units in a cypher. In short, we know what to assign significance *to.* But we do not know this about organisms. This introduces a truely radical indeterminacy into the problem: perhaps equally workable solutions can be based on incommensurable ways of individuating the physical parts and transactions to be treated.

This point is worth hammering home. I once purchased a plastic model of a computer circuit consisting, according to the directions, of three flip-flops whose interconnections could be varied in all the standard ways. By ignoring this interpretation in favor of another, it is possible to view the device as consisting of six cells, each capable of assuming eight states, whose interconnections can be varied in a great variety of nonstandard ways. Had the thing grown out of the ground in the out-back, there would evidently be no point in asking which interpretation is *right*. The only question would be, which is more useful for the purposes at hand, for instance, explaining the behavior of a large containing system. This is precisely our situation with respect to organisms.

It evidently makes sense, though complicated sense, to treat systems of flip-flops as systems of symbols whose values are determined by their states. And this allows us to make sense of the claim that such systems execute information processing programs, for when we have a rule which tells us which part to treat as which symbol, and which states count as which values, what we have is a way—though not a unique way—of translating an information processing program into one specifying physically described transactions. Given that *this* makes sense, an analogous claim about organic systems *makes sense* as well. The constraint placed on the Analytical Strategy now applies in a straightforward way, for when I have provided the translation, I have specified in physical terms the capacities to which my analysis ultimately appeals.

There is nothing irredeemably mysterious, then, about saying that organic systems execute information processing programs. But given that we *can* say this, we must ask, why anyone should *want* to, especially in the light of the two facts we have just uncovered, viz., (i) that a "translation" into a program dealing in physically specified capacities is ultimately required anyway, and (ii) that the resulting analyses will be radically *non-unique* in this sense: two completely incommensurable analyses of the same complex capacity may both be correct; two incommensurable IP programs for performing the same complex task may have adequate but different physical translations both of which are programs the organism may truely be said to execute. Indeed, this much seems demonstrable with my simple plastic computer cell.[11]

I will conclude by suggesting two rather unexciting reasons why someone might want to continue down the information processing trail in psychology in spite of these facts. First, there is a pragmatic

[11] We might reduce the indeterminacy by requiring that a "correct" information processing program be translatable into some *particular* program dealing only in physically specified capacities, e.g., the one which comes to have preferred explanatory status in physiology. This seems to render the information processing program entirely useless, but actually it does not. What the requirement amounts to is this: the preferred physiological program must have an information processing translation which accounts, *via* the Analytical Strategy, for the information processing capacities of the organism. I touch on this point again below.

motivation. The Analytical Strategy, we said, is most interesting when the difference in type and sophistication between analyzed capacity and analyzing capacities is very large, i.e., when sophisticated abilities are traded in on sophisticated organization of simple capacities. Now the mathematical theory of rule governed symbol manipulations—the theory of algorithms mainly developed by Turing—provides highly developed techniques for making such trades, provided the capacities are specified as capacities to perform symbolic operations. The prospect of promoting the capacity to store one's and zero's into the capacity to solve logic problems, speak English, recognize patterns and the like has proved irresistible as an explanatory strategy just as it has proved irresistible as an engineering strategy. The computer provides a valuable aid: as the organization gets very sophisticated it becomes difficult to tell whether we have made a successful trade. Running the program on a computer settles the matter with relative ease. So the first point is simply that the Analytical Strategy is facilitated by couching the problem in terms amenable to powerful existing analytical techniques.

My second suggestion is slightly more interesting. Some behavior is naturally described from the start in something approaching IP terms. Adding is behavior, but an adding machine is something which adds, not something which prints ink marks of such and such shapes. Some adders don't print at all, and there are various notational systems in use. Once we see this, it is clear that what makes something an adder is the fact that its behavior is subject to a certain systematic interpretation. What makes a device an adder is thus a set of facts akin to the set of facts which makes a page of cyphers an expression of a certain message. If we want to theorize about adders, and "adding behavior," therefore, we will have to abandon the vocabulary of physical capacities. Of course, every adder is some physical object or other, and each case of adding is some physical transaction or other. But there is no set of physically specifiable characteristics a thing *must* have to be an adder, though there are indefinitely many different sets of such characteristics which are sufficient. Thus, if we want to explain capacities like the capacity to add, an IP analysis recommends itself from the start precisely because it abstracts from the physical nitty-gritty. Perhaps we don't so much want to know how the little ink marks are made, but rather how it is that the device always manages to print something interpretable as a *sum*. This, it seems, requires a particular analysis of adding (chosen from the many possible analyses) and a demonstration that certain physical transactions in the device can be systematically treated as performances of the symbolic operations appealed to in that analysis. In short, we need to show that a certain IP program is executed, and to explain at the physical level why it is executed, i.e., why matters take the complex course specified. This suggests that in discussing the requirement that an information processing program be translatable into a program trafficking only in physically specified capacities we have been viewing matters from the wrong direction. Perhaps what we should say is that the (or a) preferred physical analysis of an organism's physically specified capacities must have an information processing translation which accounts, via the Analytical Strategy, for that organism's information processing capacities (i.e., those of its capacities which it can be seen to share with other organisms and devices only when described in information processing terms).

Whether psychologists should ask questions about organisms analogous to the one I raised about adders is a matter I would not presume to pronounce upon. It seems clear enough that some *do,* (especially about verbal behavior) and perhaps this is reason enough for a philosopher to subject such questions to critical scrutiny. A methodological principle I follow is that the theories scientists put forward are to be accepted more or less at face value: serious theorizing is to be presumed innocent of conceptual confusion, though not of falsehood, until proven guilty. It thus has the status of data: it is to be explained if possible and explained away only as a last resort. I have therefore presumed that programs do explain psychological capacities in some sense, the question being *what* sense. If you do not like the sense I found, and cannot find a better, perhaps you will prefer to settle for no sense at all.

APPENDIX
Execution and Representation

Suppose it granted that a capacity *c* of a device *d* can appropriately be explained by appeal to the fact that *d* executes a program *P,* execution of which amounts to or results in a manifestation of *c*. Now it is a central contention of this essay that we add nothing to the fact that *d* executes *P* by requiring that *P* be "internally represented" by *d*. It is this contention which allows me to identify the sort of explanation at issue as a species of analysis,[12] and in particular to deny that the programs of interest to psychologists are causes of the behavior they explain.[13] For if the supposition that *d* internally represents *P* adds nothing to the fact that *d* executes *P,* then whatever interest explanation by program has must derive from the fact of execution and not, as in the case of a cook following a recipe, from the alleged fact that *d* does what

[12] Specifically, the explanation is a species of functional analysis. See [4].

[13] I do not mean to deny, of course, that programs describe causes. That they do perhaps follows from the fact that the elementary instructions specify capacities explicable via the Subsumption Strategy. But the program (and the instructions) are not causes in the way a recipe is: the device does not do what it does as the effect of being so instructed.

it does *because P so instructs.* Thus, my position is summed up in the claim that the imperative sense of following instructions has no legitimate place in the sort of explanation by program favored by contemporary psychologists.

In support of this contention I offer this argument. If *d* executes *P*, then *d* represents *P:* each instruction of *P* is represented in *d* by whatever structural features of *d* account for *d*'s having the capacity specified by that instruction. Further, the "logical" organization of the instructions in *P* is mirrored in *d* by the physical organization of their representations in *d.* To see this, recall the definition of path execution in footnote twelve: were *d* to execute $s_1, \ldots, s_n$ (the first *n* instructions of a path *s* in *P*), *d* would normally execute s_{n+1}, for all *n* between zero and *m*, where *m* is the length of *s*. Associated with each s_i is some structural feature f_i of *d* which accounts, via the Subsumption Strategy, for *d*'s having the capacity c_i specified by s_i. Since *d* executes *s*, we know that were *d* to manifest $c_1, \ldots, c_n$, *d* would normally manifest c_{n+1}. That is, were f_1, ..., f_n to "do their stuff" (in that order), f_{n+1} would normally do its stuff as well, and this is exactly what must be the case if the physical organization of the f_i's is to mirror the "logical" organization of the s_i's.

This, I think, establishes that execution is sufficient for representation. But it might fairly be objected that when information processing psychologists propose to explain a behavioral capacity on the hypothesis that a program is "internally represented", they have something rather different in mind. Defenders of the view that programs are causes will, of course, insist on a distinction between a program merely being represented *by* a device, and a program being represented *for* a device. As they conceive the matter, internally represented programs constitute a kind of knowledge. The program must stand to the device which executes it much as the recipe stands to the cook. The only difference is that the recipe is "internalized": the information is *there* for use by the device in much the same way, but "there" is *inside.* This is what allows the Opposition (as I shall call them) to retain the idea that the device does what it does because instructed to do so. This is the picture Fodor draws in the following well-known passage.

> Here is the way we tie our shoes:
>
> There is a little man who lives in one's head. The little man keeps a library. When one acts upon the intention to tie one's shoes, the little man fetches down a volume entitled *Tying One's Shoes*. The volume says such things as: "Take the left free end of the shoelace in the left hand. Cross the left free end of the shoelace over the right free end of the shoelace . . ., etc."
>
> When the little man reads the instruction 'take the left free end of the shoelace in the left hand', he pushes a button on a control panel. The button is marked 'take the left free end of a shoelace in the left hand'. When depressed, it activates a series of wheels, cogs, levers, and hydraulic mechanisms. As a causal consequence of the functioning of these mechanisms, one's left hand comes to seize the appropriate end of the shoelace. Similarly, *mutatis mutandis,* for the rest of the instruction.
>
> The instructions end with the word 'end'. When the little man reads the word 'end', he returns the book of instructions to his library.
>
> That is the way we tie our shoes. ([5], p. 627)

It will be useful to have some vocabulary to help mark the difference between my conception of representation and the Opposition's conception. Let us say that *d* E-represents ('E' for 'execute') *P* if *d* represents *P* in the way I sketched two paragraphs back, i.e., if *P* is represented in *d* by the very features which account for execution. And let us say that *s* I-represents ('I' for 'information') *P*if *d* represents *P* in the sense—whatever it is—required by the Opposition view just rehearsed. My strategy is to refute the Opposition by showing that I-representation reduces to E-representation.

Consider Fodor's story about shoe tying. According to this story, the shoe tying program—call it ST—is I-represented in us by a printed volume, and we, via the little man, do whatever we do when tying our shoes because ST so instructs. Now there is no reason why we shouldn't alter this story slightly by supposing decks of punch cards in place of the books. Rather than push buttons, the little man feeds the cards into a reader which then pushes the appropriate buttons, or simply closes the appropriate circuits automatically. Again, there might as well be circuit boards instead of decks of punch cards. Rather than feed cards, the little man plugs in a circuit board labeled 'tying one's shoes'. More simply, we may suppose that all the available circuit boards are already connected to the rest of the mechanism in such a way that the little man need only throw appropriately labeled switches.

Now once the switch labeled 'tying one's shoes' is thrown, the device as a whole executes ST, and hence E-represents ST. In fact, even when that switch is not thrown, the device as a whole evidently E-represents some program *P* having ST as a subroutine: the little man executes (in the descriptive sense) the instruction 'Call ST' of *P* when he throws the switch. Indeed, we may eliminate the little man entirely from this story, and with him any hint that anything is being instructed to do anything, by supposing that whatever causes the little man to throw the switch simply causes the switch to close.

By now, all trace of a distinction between E-representation and I-representation has disappeared: the whole device E-represents—i.e., executes—a program *P* which has ST as a subroutine. There is no legitimate work to be done here by the imperative sense of 'executes ST.' The left free end of the shoelace is not taken by the left hand because the left hand (or anything else) is instructed to do so. And given the way we have generated our variation on Fodor's story, it is clear that these conclusions hold for the original story as well. What misled us was, unsurprisingly, the little man: "When the little man reads the instruction 'take the left free end of the shoelace in the left hand', he pushes a button on a control panel." Here the little man stands to ST as the cook stands to the recipe. Just as the cook stirs while heating because the recipe tells him to, so the little man pushes the button because ST tells him to. But this feature of the story is spurious. What counts is only that the buttons get pressed in the right order, i.e., in the order specified by ST, and this will happen if and only if ST is E-represented. (Imagine that the little man simply jams the book into the mechanism, which then reacts appropriately because of the book's unique *shape:* here it is obvious that the little man need not read the instructions.)

The moral is that the notion of I-representation is wrong for the same reasons that the representative theory of perception is wrong. That theory tells us that to see a pig is for a little man in our heads to perceive a representation of a pig. This is no good because little men perceiving representations of pigs is exactly on a par with big men seeing pigs. The problem is avoided if we change the theory thus: the representation is not perceived at all. It is produced and has certain effects, viz., precisely the effects which on the earlier version were supposed to follow on the little man perceiving the representation. *As theorists*, we must see certain brain patterns as representations of pigs in order to understand how pigs are seen, but the perceiver of the pig need not, nor need any of his parts. Thus, though the pig is represented, it is not a representation *to* (or *for*) the perceiver of any of his subsystems. It follows that neither he nor any of his subsystems need "understand" the "coding" in virtue of which the representation is a representation of a pig rather than a cow, or know (tacitly or explicitly) that a particular representation is a representation of a pig, or anything comparable.

Similarly, an organism which has the capacity to tie its shoes executes a shoe tying program. But it need not 'understand' that program, or know what is in it in any sense. (Beliefs may be sentences written on the brain, but the believer does not read them, nor does any of his subsystems. He is simply affected by the restructuring which we, as theorists, interpret as writing.) This contrasts strongly with the recipe case: the cook must read and understand the recipe if it is to enable him to cook the souffle.

REFERENCES

[1] Cummins, R. "Dispositions, States and Causes." *Analysis* 34 (1974): 194-204.

[2] Cummins, R. "States, Causes and the Law of Inertia." *Philosophical Studies* 29 (1976): 21-36.

[3] Cummins, R. "The Philosophical Problem of Truth of." *The Canadian Journal of Philosophy* 5 (1975): 103-122.

[4] Cummins, R. "Functional Analysis." *The Journal of Philosophy* 72 (1975): 741-765.

[5] Fodor, J. "The Appeal to Tacit Knowledge in Psychological Explanation." *The Journal of Philosophy* 65 (1968): 627-640.

[6] Newell, A., H. Simon and J. Shaw, "Elements of a Theory of Human Problem Solving." *Psychological Review* 65 (1958): 151-166.

[7] O'Shaugnessy, B. "The Powerlessness of Dispositions." *Analysis* 31 (1970): 1-15.

[8] Watson, J. *Behaviorism.* New York: Norton, 1930.