ISSUES IN THE FOUNDATIONS OF COGNITIVE PSYCHOLOGY

by

Edward Palmer Stabler, Jr.

B.A., University of Arizona
(1975)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF THE
DEGREE OF

DOCTOR OF PHILOSOPHY IN
PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1981

Signature of Author ______________________
Department of Linguistics and Philosophy
January 6, 1981

Certified by ______________________
Jerry A. Fodor
Thesis Supervisor

Accepted by ______________________
Judith J. Thomson
Chairman, Departmental Graduate Committee

ISSUES IN THE FOUNDATIONS OF COGNITIVE PSYCHOLOGY

by

EDWARD PALMER STABLER, JR.

Submitted to the Department of Philosophy
on January 6, 1981, in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy in
Philosophy

ABSTRACT

Recent work in cognitive psychology, linguistics and artificial intelligence purports to provide "computational" theories of certain "mental" states and events. This thesis develops the view that when we are careful about the construal of these theories, they are defensible and interesting, but the relation of their theoretical claims to our familiar folk psychology, on the one hand, and to computer science and engineering, on the other, is considerably less straightforward than the overlap of terminology might lead one to suppose.

Chapter One defends cognitive psychology against some a priori objections to theories that make claims about mental representations. Among other things, the classical "regress of explanation" arguments, recent versions of which can be found in the work of Daniel Dennett and Gilbert Harman, are considered.

Chapter Two considers Gilbert Ryle's suggestion that it is a mistake to think that we could ever get a theory of those mental states and events that can only come about in a certain physical or social context. This is related to what Hilary Putnam and others have called the "assumption of methodological solipsism."

Chapter Three presents a theory of computing systems based on the work of Dana Scott and others on the formal syntax and semantics of programming languages. An account is provided of what it is for a physical system to "compute a function," to "compute a program," to "use a programming language," etc. This theory is designed to have clear application to computational theories in psychology and neurophysiology.

Chapter Four makes use of the theory of Chapter Three in a consideration of the sorts of evidence that support the claim that certain rules or programs are mentally represented and govern language processing in the brain. Noam Chomsky's theory that a generative transformational grammar is mentally represented and used in langauge processing is considered in detail.

The final chapter is directed against the view proposed by Jerry Fodor, Merrill Garrett and others that the only plausible theories of language learning available are committed to the view that virtually all concepts are innate. It is argued that there is a plausible alternative that avoids this commitment.

Thesis Supervisor: Dr. Jerry A. Fodor
Title: Professor of Linguistics and Philosophy and of Psychology

ACKNOWLEDGEMENTS

This thesis could not have been written without the pioneering work done by Jerry Fodor in the philosophy of psychology and in the psychology of language. As my thesis supervisor, he has given me insightful criticism of my work and has encouraged my efforts even when my conclusions conflicted with his. I am also deeply indebted to Ned Block who provided much constructive criticism of early drafts of the thesis. James Higginbotham also read the whole manuscript and provided many helpful suggestions. I am indebted to Judy Thomson for excellent criticism of my early drafts of Chapter II. Discussions with my father, Edward Stabler, and with Jack Lee were very helpful when I was preparing Chapter III. I also had helpful discussions with Noam Chomsky about some of the issues raised in Chapters III and IV, and with Samuel Jay Keyser and Howard Lasnik about the issues raised in Chapter IV. Of course, I am solely responsible for all the errors that remain despite all the assistance I have gotten from these and other people.

TABLE OF CONTENTS

CHAPTER I

A PRIORI OBJECTIONS TO REPRESENTATIONAL THEORIES IN PSYCHOLOGY

1. Many recent theories in psychology and related fields propose explanations of mental processes according to which "mental" or "internal" representations are produced and manipulated. These theories typically suppose that organisms have mental representations that they cannot be consciously aware of; the posited representations are not introspectively accessible. Perception, for example, is thought to involve processes that we cannot be consciously aware of: information about sensory stimulation is represented and leads to the formulation of internally represented hypotheses about what is perceived. There is a long tradition of hypothesis-testing theories of discrimination and concept learning that does not generally assume that organisms are aware of the posited learning processes. (See Brown, 1974, for a review.) And the application of such theories about internal processes is not restricted to organisms that speak and understand a language. The early hypothesis-testing theories of human learning were preceded and inspired by hypothesis-testing theories of discrimination learning in rats. (Krechevsky, 1932; Levine, 1959) Such theories of animal learning are still in the running. It has

been proposed that first-language learning in pre-verbal humans also involves hypothesis-testing. (Chomsky, 1959, 1965, etc.) And even theories of the flight behavior of the common housefly maintain that houseflies internally represent visual information sufficient to enable them to compute the relative velocity of a moving target and to chase it. (Reichardt and Poggio, 1976) In this chapter a cluster of related worries about the coherence and integrity of "representational" theories like these will be considered and, hopefully, dispelled. We will not be concerned with the question of whether any of these theories are really true; rather, we will consider arguments to the effect that, regardless of the data, such theories must be incoherent of non-explanatory.

2. The basic problem

A number of different objections to these representational theories arise from a concern over how mental or internal representations can be interpreted if they are not consciously accessible. It is commonly supposed that being interpreted is what makes something a representation. C.S. Pierce(1932) says, for example, "Of course, nothing is a sign unless it is interpreted as a sign."(2.307) A sign or representation is "somthing which stands to somebody for something."(2.228) Something is a representation only in this relation to a "mind," a "scientific intelligence," an "intelligence capable of learning by experience."(2.227-2.229, 3.303, 8.176) Something like this is, I think, the natural view: something is a sign or representation only if someone takes it to stand for or express something.[1] Charles Morris says, "something is a sign only because it is interpreted as a sign of something by some interpreter."(1938, p.82)   C.I. Lewis says he agrees with Pierce that the essentials of the meaning-situation are found wherever there is anything which, for some mind, stands as a sign of something else."(1944, p.236) And Wittgenstein says that an arrow points "only in the application that a living being makes of it."(1953, §454)

These views all seem to be vaguely correct, but the truth could be stated more precisely. Is it really obvious that representations must be interpreted by some mind? In the first place, one might object that this idea has no plausibility whatever as a view about representation-tokens, i.e., as a view about particular utterances or inscriptions of representations. A book that waits at the bookstore for its first reader is, one would think, full of representation-tokens that have never even been percieved, let alone rercognized and taken to stand for or express something. So perhaps the view about representation that

---

[1] Pierce intends these views to apply to *all* signs, but some philosophers have argued that "natural signs" need not be interpreted. Allston(1964, ch.3) argues, for example, that boulders of a certain kind are a sign of glacial activity and that "they would still *be* signs of glacial activity, even if no one should ever realize this." So he distinguishes the notion of *x being taken as a sign of y* from the notion of *x being a sign of y*; he argues that only the former notion presupposes that some mind is involved. If Allston is right about this, then the discussion in this chapter is to be understood as concerning only the former notion. It is the former notion that is relevant to the accounts we will consider here, as well as to computational accounts of physical systems in general.

we are after should be that if anything is a representation-token, then either it or some other token of the same type must have been interpreted as such by some mind.

But even this view does not pass preliminary inspection. There are also representation-types that are perfectly meaningful even though no tokens of those types have ever been interpreted. There are surely indefinitely many English sentence-types, for example, of which no token has ever been interpreted by any mind. A token of such a sentence-type may be a meaningful representation-token despite the fact that no one has ever interpreted a sentence of that type. If I had meant to write 'The sun was shining on the sea' but accidentally wrote 'The sun was shining on the pea' without noticing my error, then my manuscript might well contain a sentence-token of a type that had never been interpreted by anyone before, but it would be a meaningful sentence nevertheless. Its interpretation as a sentence of the language is determined. Similarly, for a system of analog representation, we might have rules of interpretation that specify an interpretation for infinitely many representation-types, despite that fact that there will always be representation-types that have never been considered at all by any mind. If this is correct, then we need to modify the proposed view about representation. Let's say that a representation-type is interpreted if either a token of that type has been interpreted by some mind, or if its interpretation as part of some system of representation (used by some mind) is determined, as in these cases. Then we can say that something is a representation-token only if *either* it has been interpreted as a representation by someone *or* it is of a type that has been interpreted.

Before considering whether this view is going to expose a problem in theories of mental representation, there is one other sort of case that should be taken into account, viz., representations which are produced and used by machines of one sort or another. A tape recorder produces a representation of its acoustic input, and a computer may store a representation of the trajectory of a missile. Are these representations interpreted? Surely we want to say that they are. One might say that these representations are taken to be representations not by humans but by the machines themselves. But this position really seems implausible, at least for machines like tape recorders, calculators and typical computers. It is absurd to suppose that the tape recorder itself takes the pattern of magnetization on its tape to represent, as it might be, Jimmy Carter speaking about nuclear disarmament; it is we humans who recognize what is represented on the tape. It seems equally absurd to suppose that a computer (or a computer program) takes the states of its memory devices to represent anything; it is the human mind that recognizes that certain states can be systematically interpreted as it might be, numerically, and that certain state transitions can then be described as computation. So until some good argument to the contrary is presented, let's assume that the representations produced by such machines *are* interpreted; they are interpreted by us, by people. It is often very useful to so interpret the states of machines or of their associated equipment when we want to give an account of their operation. In the typical case we do not see or hear representation-tokens of this sort, but nevertheless we may know perfectly well that they exist and that they play a role in the operation of these machines. So it looks like these cases provide no reason to modify the proposed account of representation: something is a representation only if either it has been interpreted as such by someone or if it is of a type that has been intepreted.

Now let's consider whether this account of representation is going to cause problems for representational psychological theories. Consider theories that claim that organisms have internal representations that they are not aware of. If the organisms are not even aware of the internal representations, then how can they be interpreted? Clearly the answer is that they are not interpreted by the organism in such cases, but by the theorist. The theorist does not need to assume that the subject interprets his own internal representations any more than we need to assume that a tape recorder or computer itself takes its representations to stand for something. The psychologist has no problem unless he claims both that a subject has internal representations he is not aware of and that the subject himself interprets them as such. Surely no psychological theory is committed to both of these claims.

3. Dennett on the basic problem

The basic problem is to give a satisfactory account of the interpretation of internal representations that does not require that the subject be aware of them. An account different from the one just presented has been offerred by Daniel Dennett in a recent series of papers, but his alternative account seems incoherent. On the one hand, he urges that views like the one defended here are "misconceived;" they are misconceived because they do not acknowledge "the important distinction between the content of a signal to the system which it informs, and the content we on the outside can assign it when we describe the signal and the system of which it is a part."(1977, p.100) He suggests that the latter notion of content is not the one the psychologists are using, so the relevant notion of content must be the former, the content of the signal "to the system it informs." Dennett calls the internal mechanisms that interpret and use these internal representations "inner exempt agents." But then, on the other hand, he claims that computer science has given us an important conceptual advance in showing that internal representations are "vehicles of representation that function _without_ exempt agents."(1972, p.102; cf. also 1978, p.123) So the internal representations are not interpreted by internal interpreters "on the inside" after all. Since they are not interpreted by us "on the outside either, we seem to have a _reductio ad absurdum_ of Dennett's views. Surely it is absurd to suppose that there are internal representations even though nothing internal or external interprets them. But Dennett is apparently willing to accept this uncomfortable result. He says that internal representations are "self-understanding," but never explains what that means. Why does he accept this seemingly incoherent position?

To begin with, Dennett ultimately wants to reject the view that internal representations require internal understanders which interpret and use them, "inner exempt agents." This is certainly right; the requirement of internal understanders is untenable. Dennett suggests that this requirement brings the threat of an infinite regress of explanation, and this view will be discussed below. But the requirement also seems to be faced at the outset with counterexamples like the ones mentioned above. A tape recorder does not understand that the representations on its tape represent anything, and the representations could still be produced even if the tape recorder had no mechanism for even playing back what it had recorded. In this case we would have representations that are neither understood by an internal mechanism nor used at all. A computer or calculator does not take its states to stand for numeric values; the user "on the outside" does. It is useful to interpret calculator memory states numerically because its changes of state may then correspond to numerical operations that we would like to have performed.[2] <u>This</u> is the basis of a symbolic, computational

[2] A theory of computation is presented in Chapter III which spells out in detail what is involved in giving such an account of calculators and other computing machines.

account of the role of the representations. It is a mistake to suppose that there are any internal understanders who "read" the internal representations and act accordingly. The interpreted states just have a certain role; they have a certain effect on the operation of the machine such that memory state transitions correspond to numerical computations. So we can agree with Dennett that internal representations do not need internal understanders.

The problem is that Dennett rejects not only internal interpreters but also external ones, so he is apparently saddled with the view that there are no interpreters at all. He notes that this result might make some people uncomfortable:

> ...one could insist that the very lack of exempt agents in computers to be users of the putative representations shows that computers do not contain representations -- real representations -- at all, but unless one views this as a rather modest bit of lexicographical purism, one is in danger of discarding one of the most promising conceptual advances ever to fall into philosophers' hands. (1977, p.102)

The worry about the missing interpreters, the missing "exempt agents," is not just a lexicographical concern, however; it is a worry about the very coherence of the account, no matter what terms it is expressed in. If we grant that the internal states of a computer or of an organism may be taken to represent things, then it is not mere lexicographical purism that prompts one to inquire, "By whom, or by what, are these states taken to stand for or to express something?" It is not mere lexicographical purism that makes the answer, "By nothing internal and nothing external" unsatisfactory. If no one and no thing takes these states to represent anything, to stand for anything, or to express anything, then surely they don't.

Why does Dennett reject the view that internal representations are interpreted by the theorist "on the outside? As was noted above, he suggests that the view is misconceived, but the reasons for this suggestion are not made entirely clear. He urges that imposing our interpretation of internal states fro the outside does not provide the interpretation the psychologist needs. He says,

> For instance, one badly misconceives the problem of perception if one views the retinal receptors as "telling" the first level of hypotheses testers "red wavelength at location L again," for that level does no utilize or understand (in any impoverished sense) information of that sort. (1977, p.100)

The point of this example is apparently to show that the theoris can interpret internal states and events in a way that is not appropriate for a theoretical account of the operation of the system. We on the outside might recognize that a particular signal can be taken to represent red light at L even when, as fa as the system is concerned, precisely the same effect might have been produced by a signal resulting from green light at location M. In this situation we might say that the system does not "utilize or understand" our interpretation of these signals, because it does not respond differently to the two cases. But this suggests only that our interpretation of the signals is not a theoretically appropriate one; since the theorist would presumably want to give the same account of both cases, he would not want an interpretation that distinguishes them. So, for theoretical purposes, the choice of an interpretation will be constrained by such facts. But this does not show that the theoretically relevant interpretation is not an interpretation imposed by the theorist at all. Surely there is no reason here for supposing that the system has its own interpretation of its states and events; the theorist is the only being who actually recognizes any content in the neural signals.

Another example that Dennett uses to illustrate the importance of distinguishing the content of a representation "to the system" is a little clearer. He considers the possibility of discovering that "certain features of brain activity could be interpreted as a code," and he says,

> Discovering such a code is not establishing that the information the code carries is also carried for the person or even for his brain. D.H. Perkel and T.H. Bullock...discuss the discovery of a code "carrying" phasic information about wing position in a locust; it is accurately coded, but "the insect apparently makes no use of this information." (Blocking this input and substituting random input produces no loss of flying rhythm, ability, etc.) (1971, p.43n)

Once again, though, the example does not support Dennett's conclusion that the system has its own interpretation of its internal representations. The fact that we can recognize some representations of information that play no role in the functioning of the organism does not support the conclusion that we are not the interpreters of those representations that do play a role. Quite to the contrary -- it is not hard to see why psychologists, like computer scientists, would be particularly interested in representations that play a significant, specifiable role in the operation of the system under study. It seems that Perkel and Bullock express their dicovery appropriately in saying that they had discovered the representations of information that the insect does not use. There can be representations of information that is not utilized at all by the system, as well as representations that play significant roles; both can be recognized by the theorist. The point is that only the latter representations wil be significant for a theory of the functioning of the organism.

It seems that this is the sort of point that Dennett is really after, and that it can be accepted without the absurd commitment to uninterpreted representation. This is, I think, the truth that underlies Dennett's remark that "What makes it the case ultimately that something in this sense represents something within a system is that it has a function within the system...Content is a function of function...."(1977, p.106) This remark is misleading in its suggestion that having a function in the system is what makes something a representation; surely if it is not interpreted by anyone or anything it is not a representation whether it has a function or not. The remark also appears to rule out cases like the one just discussed in which we can recognize representations of information that have no function in the system. The correct point in Dennett's claim is, I think, just that psychologists are (typically) interested in using interpretations of internal states and events that lend themselves to giving a general theoretical account of the functioning of the system. Thus even if the theoretically relevant interpretation of an internal state or event is not a "*function* of function," it will at least be "*related* to function."[3] We could give an alphabetic interpretation to memory states of a numerical caluculator, but that would not be expedient for an account of the system's operation. Or, as in the first example, we could specify an interpretation of internal states and events that would not be appropriate for all instances of the theoretically relevant kind.[4] The appropriate theoretical interpretaion will be sharply constrained. Still, we can maintain that the interpreted internal states are interpreted *by the theorists*, by humans on the outside of the systems. This position has the enormous advantage of avoiding the commitment to representations that are taken to have content by no one and no thing at all. That is, this position avoids the *reductio ad absurdum* that undermines Dennett's position. We can relapse into the natural position with which we started: representations that are produced and used by machines and organisms are interpreted by other humans.

----

[3] It should be noted that some psychologists want to claim that "metal representations" have a determinate content, that we are not free to interpret them one way or another as we might the states of a calculator. This view is not, or at least not obviously, incompatible with the view defended here. It could be that in some cases the causal role of a state -- the way it is related to inputs, outputs and other states -- determines completely the theoretically relevant, correct, interpretation of the state, and this may be all the psyhcologist needs. In such a case, the content of the representations would be, as it were, a "*function* of function." See, e.g., Loar(ms), and Fodor(ms) for discussions of the determinacy of mental representations.

[4] This sort of point is discussed in Chapter II and in Fodor, 1980.

4. Related problems

There are other challenges to representational theories in psychology that are very similar to the basic problem we have been considering. The basic problem concerns the interpretation of representations; one collection of related problems concerns the understanding of a language. Norman Malcolm points out in a discussion of the neurophysiological basis of memory that symbols, rules and descriptions are used by people who understand a langauge. He asks, "How does the brain of a person or animal fit into this picture?...Does it apply words or other "symbols" to objects and situations" (1977, pp.208-209) In a criticism of a theory of first language learning that attributes internal representations of rules to infants, Gilbert Harman says, "It does not seem to make sense to suppose that [someone] can represent rules without representing them in some language," but certainly we do not want to accept the "absurd assumption that before he learned his first language [the language learner] already knew another language." (1967, pp.76, 77) This is to "treat the child as if he were a linguist investigating some hitherto unknown language." (1968, p.661) In these passages Malcolm and Harman are worried about a feature of psychological theories that is really quite common: the attribution of representations in a language to things that do not understand a language. Other theories apparently assume that rats use a language in which they couch their hypotheses about discrimination learning situations, and others, that houseflies use a language in which they can carry out the necessary computations of flight trajectory. The problem, then, is to explain how any of these theories could be reasonable. Isn't it obviously a mistake to attribute "languages," systems of representation, to organisms that apparently do not understand any language?

This problem is just the "basic problem" in a new guise; our resolution of the basic problem provides the basis for the resolution of this one as well. Suppose we make a tape recording of a speech that includes descriptions, rules and hypotheses. The tape recorder then has representations of descriptions, rules and hypotheses on its tape, representations that play a role in generating the acoustic output we get when the tape is played back. There is no problem with this account of the operation of the tape recorder because there is absolutely no need to assume that the tape recorder itself understands the language it has represented. We use <u>our</u> interpretation of the states of the tape to provide this convenient account of what is going on.

The situation is essentially the same for computers. a computer can also store and manipulate the text of a speech in English. There is no need to assume that the computer understands English in such a case. We are the ones who understand the internal states of the machine as a representation of the text, and so we can describe changes of state as the "manipulation" of the text. Similarly, we might interpret states of the computer as the representation of a hypothesis and interpret certain input to the computer as data relevant to the hypothesis. In this case we could describe the operation of the computer as hypothesis testing if certain inputs would cause appropriate changes in the states that represent the hypotheses.

Computers (and sometimes also computer programs) are said to _use_ certain programming languages, and even sometimes to _understand_ them. There is nothing wrong with saying this, perhaps, so long as it is recognized as an unnecessary bit of anthropomorphism. That is, we do not _need_ to assume that the computer (or the program) is itself recognizing the representational content of the representations that _we_ recognize the machine to have. A computer is using (and "understands") a programming language just in case there is some interpretation of its states under which formulae of that language play a certain role in the operation of the computer. (See Chapter III for a detailed account.) the computer does not "read" and interpret the representations and then act accordingly; again, the interpreted states simply have a certain effect on the operation of the operation of the machine which can be given a symbolic description. We do not need to assume that anyone or anything other than we humans recognizes the interpretation of the states or "understands" the programming language in the sense of taking formulae of the language to stand for or express something. Someone might want to argue for the view that in some cases a computer (or its program) also "understands" a language in this latter sense, in the way we do, but this view seems entirely implausible at least for the machines and programs we have now.

In any case, there are clear cases where we correctly attribute to a machine representations in a language that the machine does not understand, as in the case of computers and tape recorders with representations in English. Thus, the mere attribution of internal representations in a language does not commit us to the view that the system under study understands or interprets that language. Representational theories need not have any commitment to internal understanders or interpreters.

5. The regress arguments

The problems discussed so far have also been associated with some "regress arguments" against certain psychological theories. Getting straight about the basic problem and its near relatives unfortunately does not defuse the regress arguments entirely, so they deserve separate consideration. Dennett has argued that the assumption that internal representations require internal interpreters, "inner exempt agents," would (if it were accepted) threaten representational theories in psychology with the danger of an infinite regress. He says,

> The only psychology that could possibly succeed is neo-cognitivist, which requires the postulation of an internal system of representations. However, nothing is intrinsically a representation of anything; something is a representation only for or to someone; any representation or system of representations requires at least one user of the system who is external to the system. Call such a user an exempt agent. Hence, in addition to a system of internal representations, neo-cognitivism requires the postulation of an inner exempt agent or agents -- in short, undischarged homunculi. Any psychology with undischarged homunculi is doomed to circularity or infinite regress, hence psychology is impossible.... (1977, p.101)

This passage leaves a number of points unclear. _What_ circularity and _what_ regress is psychology doomed to? And _why_ is a representational psycholgy doomed to one or the other of these fates? Apparently Dennett has something like the following argument in mind:

> (1) If a psychological theory attributes internal representations to an organism, then it is committed to some internal being or mechanism that interprets and uses them.
>
> (2) Presumably our explanation of any such internal interpreter will involve attributing internal representations to _it_, and these must also be interpreted by some internal interpreter that will also need to be explained, and so on.
>
> (3) Continuing the account in this way will generate an infinite regress of explanations of an infinite number of internal interpreters, unless the account is circular (i.e., circular in that the system of internal representations used by some internal interpreter _I_ is interpreted either by _I_ itself or by an interpreter at some "higher" level whose own internal internal representations _I_ is ultimately involved in interpreting).
>
> (4) Any explanation that leads to such an infinite regress or to such circularity must be incorrect.

Let's call this the "regress of interpreters" argument. There are other arguments of basically the same form. Dennett notes the follwing examples:

> For instance, it seems (to many) that we cannot account for perception unless we suppose it provides us with an internal image (or model or map) of the external world, and yet what good would that image do us unless we have an inner eye to perceive it, and how are we to explain its capacity for perception? It also seems (to many) that understanding a heard sentence must be somehow translating it into some internal message, but how will this message in turn be understood: by translating it into something else? (1978, p.122)

Here we have the beginnings of a "regress of perceivers" and of a "regress of understanders." The threat of a "regress of understanders" has been used by Gilbert Harman in a similar argument against Noam Chomsky's theory of language understanding:

> Taken literally, he would be saying that we are to explain how it is that Smith knows how to speak and understand a language by citing his knowledge of another more basic language in which he has (unconsciously) "internally represented" the rules of the first language. (It does not seem to make sense to assume that Smith can represent rules without representing them in some language.) The main problem with such a literal interpretation of these remarks would be the implausibility of the resulting view. How, for example, would Smith understand the more basic language? In order to avoid either an infinite regress or a vicious circle, one would have to suppose that Smith can understand at least one language directly, without unconsciously knowing the rules for that language. But if this is admitted, there is no reason why Smith cannot know directly the language he speaks. Thus, literally interpreted, Chomsky's theory would almost certainly be false. (1967, p.76)

It will be argued that all of these regress arguments are fallacious. They provide no good reason for thinking there is a problem with the theories against which they are directed.

Consider the "regress of interpreters" argument. Its first premise has already been challenged, but its second premise is also a mistake. It is a mistake to see a regress getting started in any of the theories considered here. Why would anyone accept premise (2) of the "regress of interpreters" argument? Well, it is forced upon anyone who accepts both the first premise and the general claim that all "interpreters" use internal representations. But surely no psychological theory explicitly asserts both of these claims.[5] Typically the first is explicit, and it is the critic who presumes a commitment to the general claim. The passage by Harman suggests the most likely basis for such a presumption on the part of the critic, viz., that the very reasons for proposing the "first level" of internal representation are present at every level, so if a further level of representation is not needed at some point then there is no reason to suppose that it was needed in the first place to explain the original phenomena.

Suppose, for example, that the "regress of interpreters" argument is directed against a theory of language understandidng or some other theory about what is involved in the interpretation of perceived symbols. If such a theory posits a system of internal representation to explain the interpretation of the

---

[5] Some regress arguments are clearly directed against views like these in which there is some general claim about how something is to be explained. Cf., e.g., Ryle, 1949, pp.28-32; Block, 1979, pp.4-5. These arguments do not, I think, apply to any psychological theory that has been seriously proposed.

symbols, then it is natural to presume that we would want to posit a further system of representation to explain the interpretation of the posited representations; after all, the "interpretation" of symbols is what needs to be explained in both cases. If, on the other hand,we do not invoke a second level of internal representation but can successfully explain how the first internal representations simply play some role R in certain internal processes, then this would show the first level of internal representation to be unnecessary; the original phenomena could then be explained by supposing that the perceived symbols play some role R', similar to or identical to R, in internal processes. Something like this is apparently the idea behind the proposed defense of premise (2). But it is easy to see that this defense is not generally going to work.

Consider a typical representational theory of language understanding which presumes that the posited internal representations are neurophysiological states under some interpretation, and which provides some preliminary account of their role in internal processing and in the causation of behavior that exhibits an understanding of what has been understood. It is certainly far from clear how such a theory could be recast so as to avoid the commitment to internal representations even if we did have an account of the internal processes involving the internal representations which did not attribute a further level of representation. It is wildly implausible that the symbols perceived, the acoustic signals or whatever, are themselves accessed by the internal processes. The symbols perceived surely do not "act at a (spatial and temporal) distance" to influence internal processes directly. If this hypothesis is even metaphysically possible, it could presumably be defeated experimentally by showing that the influence of the external symbols depends entirely on the mediation of certain internal states as the original theory claims. So, in sum, there _is_ a difference between the original phenomena of language understanding and the posited internal phenomena, even if the problem in both cases is to explain "interpretation;" the obvious difference between to two cases provides clear reason for supposing that the internal mechanisms do _not_ call for the same sort of explanation as is given of the original phenomena. Therefore, it is a mistake to see any danger of an infinite regress here, regardless of whether the internal mechanisms ought to be called "interpreters" or not.

The other regress arguments suffer from analogous problems. As a last example, let's consider Harman's argument against Chomsky's theory that a subject uses "unconscious" internally represented grammatical rules in understanding his language. (The argument is presented in the passage quoted above.) Harman is apparently assuming that if the internally represented rules are represented in a language then the subject must understand that language. This assumption was challenged above (in §4), but let's set this issue aside. Suppose that the rules _are_ somehow "understood" by the subject, or rather, by internal mechanisms in the subject which have access to the internal representations. Then the question is: how is this internal language understood?

And, more generally, what influence do the internal representations have on internal processes?

As a matter of fact, Chomsky does not attempt to answer this question. He does not offer any account of what role the internal representations play in in the internal processing. The whole question is still substantially uninvestigated, a matter for speculation. Certainly it is not clear that the subject must have a further system of grammatical rules for understanding the internal language in which the first rules are represented, and a further set of rules for understanding the second set, and so on. There are good reasons for thinking that the "understanding" of the internal language will need an explanation that is substantially different from the propsed explanation of the understanding of the spoken language. In the first place, as in the last example, there is good reason to suppose that the internal representations can play a role in internal processing that the representation-tokens of the spoken language cannot play. And in the second place, the internal language may be quite different from the spoken language. Thus we can grant Harman's point that some langauge must be understood "directly", i.e., without the representation of its grammatical rules in some other language, and still have good reason to reject the view that there is then "no reason why Smith cannot know directly the language he speaks." Harman apparently just presumes, without defense, that the appropriate explanation of natural language understanding will also be the appropriate explanation of internal language understanding, and vice versa. This presumption seems most implausible and in need of defense, yet it is mustered without defense as the basis fo the conclusion that Chomsky's theory, literally interpreted, is almost certainly false.

The basic idea behind the regress arguments is that the explanations at which they are directed do not get us anywhere; they move the original problem into the head or into the mysterious realm of the mental, leaving it essentially unanswered at that level. Since we then are left with the same sort of problem inside the head, presumably we want to offer the same sort of explanation for it, and carrying on in this way generates an infinite regress of explanations. Our response to this idea has been to point out that in each case the representational theories do not leave us with the same problems that they set out to explain in the first place. On the contrary, they leave us with different problems, problems which typically do not call for the same sort of explanation as the original problem called for. They leave us with different problems because they do get us somewhere, that is, they do have content, and in the normal course of scientific inquiry this content is tested and confirmed by relevant data. The explanations of the internal mechanisms posited by representational theories are typically quite different from the explanations of the original phenomena, but, in any case, these will also have to account for the data and survive whatever testing can be brought to bear.

There is a tendency, I think, in some responses to the regress arguments to concede too much. Many of the recent discussions of these arguments mention the computational accounts of the internal mechanisms posited by representational theories, accounts founded in our recently acquired understanding of the operation of computers.[6] These accounts are of considerable interest, but it does not seem to me that they ought to be even mentioned in this context. The methodological respectability and empirical plausibility of the representational theories does not hang on the success of these accounts of the internal processes. If these new computational theories turn out to be inadequate, that would leave psychologists with the difficult problem of finding some other sort of account of the posited internal processes, but it would not in itself indicate any defect in the representational theories. Even in the physical sciences it is common to invoke mechanisms for which no adequate account is to be had. The famous example (whose time may at last be passing) is our inability to explain the gravitational forces invoked by Newton and his successors; the problem of explaining gravitation remained, yet the physics developed and has become the paradigm of good science. It could happen that the representational theories in psychology will also continue to be developed and refined, that the domain of phenomena that they can explain will be extended dramatically, without an adequate account of the posited internal mechanisms. In short, the regress arguments do

----

[6] See, e.g., Fodor, 1968; Chomsky, 1969; Dennett, 1975a, 1977, 1978; Rorty, 1977; Block, 1979.

not suffice to show any flaw in representational theories. The arguments that the theories are committed to infinite regresses are fallacious, and if they do succeed at least in pointing out problems that remain unexplained, that does not show the theories to be any worse off than the best. To suppose that we <u>need</u> the new explanations to answer the regress arguments is to concede too much to them.

6. Related objections

There is another objection to representational theories that is sometimes associated with what we have called "the basic problem" and with the regress arguments. This objection should be mentioned here, though a detailed consideration of it goes well beyond the scope of this chapter. Consider the following passages from Norman Malcolm(1977) on theories of memory:

> If the man who went into he next room to fetch a bolt of cloth to match the color of the one he had just seen carried in his mind an image of that color, then (we think) he would know which color to fetch, and there would be no mystery about his getting it right...There seems to be a knowledge-gap between the man's looking at the first bolt and his subsequent response of selecting in the next room another bolt of the <u>right</u> color. We felt the need to fill this gap with an image (attended with certain feelings) that he consulted as he selected the second bolt. But when we think about this intermediary, we realize that it, too, presents us with a gap. For why does an image that feels familiar <u>guide</u> the man's action? How does <u>it</u> inform him which cloth is of the right color? Surely we have another gap here that needs to be filled with a second intermediary whose function will be to inform the man how to interpret the first intermediary. But exactly the same sort of question can be raised about the second intermediary. What tells the man how to interpret <u>it</u>? Thus we are confronted with still another gap, which requires still another intermediary, and so on! (pp.91-92)

The conclusion that should be drawn is that the thinking up of possible features of mental content will bring us

> no nearer to the goal of understanding memory. No hypothesis about what goes on in our minds when we remember, as to how remembering works, will remove our puzzlement. (p.93)

It is this last conclusion I am interested in: the view that "No hypothesis...will remove our puzzlement." The first thing to notice is that this conclusion is not supported by the regress argument. In the first place, the regress argument has no application at all to theories that do not posit images or other representations, yet the conclusion is that no theory will suffice. And in the second place, the regress argument is fallacious. The subject does not need to interpret his own memory images, at least on many accounts, and the conclusion is not restricted to those that do have this requirement. And we do not need further representations to account for an image's role in the processes leading to the subject's response, so we are not driven into a regress. The regress argument provides no reason for thinking that we will not be able to get a good representational theory.

The interesting point is that some of Malcolm's remarks suggest that even if we actually had a complete account of all the internal processes "involved" in remembering, he would still want to maintain that this theory would not explain how we remember. He suggests that this is really not something that calls for any explanation at all: "...why are we not content to hold that a person's ability to give such a report [of something he witnessed earlier] is a primitive fact about people? Why are we not willing to allow that this is a natural human power?"

(p.89; cf. also pp.101-102)

This sort of view cannot be supported with a regress argument, though there may be other reasons for accepting it.[7] In particular cases, for example, by taking a certain line on the nature of mental events like remembering, it might be argued that a particular theory does not explain any such events. Suppose that I see a bolt of cloth, and that later on I remember it and tell you what color it was. Imagine a theory that provided grounds for roughly the following sort of account: Light reflected from the cloth produced an image on the retinas of my eyes; this produced an internal representation of certain information about the cloth (including its color) which was stored in memory; later, some of this represented information was retrieved by the processes leading to my verbal report of the color of the cloth. Now, someone might want to argue that this is just an account of certain neurophysiological processes under a computational interpretation, and as such it cannot explain how I remembered the color of the cloth. No one doubts that certain things were going on in my nervous system and that these things were responsible for my articulation of the report. What _can_ be doubted is that any account of these things is (or provides grounds for) an explanation of _how I remembered_. Of course, if my remembering _is_ just a certain sort of retrieval of stored information that occurs in my brain, then this theory presumably

[7] A different argument for the view that no scientific theory will be able to explain certain mental events is considered in the following chapter.

could provide an explanation of my remembering, but just such identity claims are currently a subject of controversy. It is at least not obvious that any computational theory can provide an explanation of my remembering.

This sort of claim is interesting, but it is not an a priori objection to representational theories in psychology as the other arguments were. That is, it is not an argument that purports to show representational theories to be confused or devoid of content; rather, it is a claim about what such theories can be expected to explain. On any account, the things that so-called theories of memory do explain are going to be intimately related to remembering, but nevertheless the question of whether they can actually explain the remembering itself is of interest. Unfortunately, it is not a question which will be explored here.

BIBLIOGRAPHY FOR CHAPTER I

Allston, W.P. (1964) *Philosophy of Language*. Englewood Cliffs, New jersey: Prentice-Hall.

Atherton, M. and Schwartz, R. (1974) Linguistic innateness and its evidence. *The Journal of Philosophy*, *71*, pp.155-168.

Block, N. (1979) What is philosophy of psychology? Reprinted in N. Block, ed., *Readings in the Philosophy of Psychology, Volume 1*. Cambridge, Massachusetts: Harvard University Press, 1980, pp.1-8.

Brown, A.S. (1974) Examination of the hypothesis-sampling theory. *Psychological Bulletin*, *81*, pp.773-790.

Chomsky, N. (1959) Review of Skinner's *Verbal Behavior*. *Language*, *35*, pp.26-58.

Chomsky, N. (1965) *Aspects of the Theory of Syntax*. Cambridge, Massachusetts: MIT Press.

Chomsky, N. (1969) Linguistics and philosophy. In S. Hook, ed., *Language and Philosophy*. New York: New York University Press.

Dennett, D. (1969) *Content and Consciousness*. New York: Humanities Press.

Dennett, D. (1975) Brain writing and mind reading. Reprinted in Dennett, 1978a, pp.39-50.

Dennett, D. (1975a) Why the law of effect will not go away. Reprinted in Dennett, 1978a, pp.71-89.

Dennett, D. (1977) A cure for the common code? Reprinted in Dennett, 1978a, pp.90-108.

Dennett, D. (1978) Artificial intelligence as philosophy and psychology. Reprinted in Dennett, 1978a, pp.109-126.

Dennett, D. (1978a) *Brainstorms: Philosophical Essays on Mind and Psychology*. Montgomery, Vermont: Bradford Books.

Fodor, J.A. (1968) The appeal to tacit knowledge in psychological explanation. *The Journal of Philosophy*, *65*, pp.627-640.

Fodor, J.A. (1975) *The Language of Thought*. New York: Crowell.

Fodor, J.A. (1980) Methodological solipsism considered as a research strategy in cognitive psychology. *The Behavioral and Brain Sciences*, *3*, pp.63-109.

Fodor, J.A. (ms) The mind-body problem. *Scientific American*, forthcoming.

Fodor, J.A. and Pylyshyn, Z. (ms) How direct is visual perception? Forthcoming.

Gregory, R.L. (1970) *The Intelligent Eye*. New York: McGraw-Hill.

Gregory, R.L. (1974) Perceptions as hypotheses. In S.C. Brown, ed., *Philosophy of Psychology*. New York: Barnes and Noble.

Harman, G. (1967) Psychological aspects of syntax. *The Journal of Philosophy*, *64*, pp.75-87.

Harman, G. (1968) Three levels of meaning. *The Journal of Philosophy*, *65*, pp.590-602.

Krechevsky, I. (1932) "Hypotheses" in rats. *Psychological Review*, *39*, pp.516-532.

Levine, M. (1959) A model of hypothesis behavior in discrimination learning set. *Psychological Review*, *66*, pp.353-366.

Lewis, C.I. (1944) The modes of meaning. *Philosophy and Phenomenological Research*, *4*, pp.236-249.

Loar, B. (ms) Mind and Meaning. Cambridge: Cambridge University Press, forthcoming.

Malcolm, N. (1977) Memory and Mind. Ithaca, New York: Cornell University Press.

Morris, C. (1938) Foundations of the theory of signs. Foundations of the Unity of Science, 1(2), pp.77-137.

Pierce, C.S. (1932) Collected Papers of Charles Sanders Pierce. Edited by C. Hartshorne and P. Weiss. Cambridge, Massachusetts: Harvard University Press.

Reichardt, W. and Poggio, T. (1976) Visual control of orientation behavior in the fly. Quarterly Review of Biophysics, 9, pp.311-375.

Restle, F. (1975) Learning: Animal Behavior and Human Cognition. New York: McGraw-Hill.

Rorty, R. (1977) Wittgensteinian philosophy and empirical psychology. Philosophical Studies, 31, pp.151-172.

Ryle, G. (1949) The Concept of Mind. New York: Barnes and Noble.

CHAPTER II

MENTAL TERMS IN PSYCHOLOGICAL THEORY:
METHODOLOGICAL SOLIPSISM AND RELATED CONSTRAINTS

1. In Chapter VII of Dilemmas(1954) Gilbert Ryle argues that there is a "strong pressure" to accept the "mistaken assumption" that "seeing, hearing and the rest" are states or processes that scientists can investigate, and that this assumption is a source of a problem about how we can percieve the external world. He says, "The programme...of locating, inspecting and measuring the process or state of seeing, and of correlating it with other states and processes is a hopeless programme...."(p. 104) In accepting this mistaken assumption, he says,

> ...we have yielded to the temptation to push the concepts of seeing, hearing and the rest through the hoops that are the proper ones for the concepts which belong to the sciences of optics, acoustics, physiology and psychology. The unscheduled but well-disciplined conduct in ratiocination of the notions of seeing, hearing and the rest diverges sharply from the conduct we have been induced to schedule for them.(pp.109-110)

In support of this view, Ryle urges that "no one would ever suppose that 'winning' stood for a physiological or psychological condition or process," and that certain mental terms like 'seeing' are analogous to 'winning' in important respects. We would not expect physiologists or psychologists to be able to develop a scientific theory of winning, and similarly, we should not expect to be able to develop a theory of seeing or hearing.

One of the important analogies between winning and seeing that Ryle mentions here is that neither sort of event occurs merely in virtue of what goes on inside of the subject. Thus he says,

> A runner's victory, though it is tied up, in lots of important ways, with his muscles, nerves and frame of mind, with his early training and briefing recieved just before the race, still refuses to be listed among these kindred phases of his private career. However fast, resolutely and cleverly he has run, he has not won the race unless he had at least on rival, did not cheat and got to the tape first. That these conditions were satisfied cannot be ascertained by probing still further into him.(pp.105-106)

In short, winning a race is the sort of thing that can occur only in a certain context. And similarly, contextual conditions must be satisfied for one to see something. Ryle illustrates this point in the following passage:

> ...the question whether or not the spectators saw the doves emerging from the conjuror's pocket is for him, not them, to decide. Notice that he is in a position to reject the claim of the spectators that they saw it happen, if he knows it did not happen. But if they claim to have seen something happen which did happen, then he cannot, on this score alone, concede their claim. If the thing happened, but happened behind a screen, then their claim to have seen it must be rejected. They could not have seen it unless it happened, and unless it happened in such a place, and at such a distance and in such a light that it was visible to them and unless their eyes were open, properly directed and so on.(pp.107-108)

So Ryle is apparently claiming that seeing doves emerge from a pocket is also the sort of thing that can occur only in a certain context, and that this is one reason for thinking that it is not a matter for scientific investigation. The question I want to consider in this chapter is whether there is any sound argument here. Granting that 'seeing a dove' is truly applicable only in a certain sort of context, what follows with regard to the use of this term in psychology?

Ryle suggests that 'seeing', 'hearing' and the rest are unlike the terms of optics, acoustics, and physiology in requiring a certain context for their true application. But even if he is right that the terms of the physical sciences are different in this way, we want to know why this is important. Why shouldn't the physical sciences use terms that have contextual conditions on their application? Now this is a question that we can answer, for it is apparent that to have certain conditions on the application of some physical terms would render them unsuitable for scientific theory, and it could be that the conditions on the application of certain mental terms are analogous to these unsuitable ones in crucial respects. This will be the strategy then: to characterize a kind of term that is unsuitable in a scientific theory, to point out the problems that may be engendered by the use of such terms, and to point out that, in certain domains, some ordinary, pre-theoretical terms are of this kind.

2. One sort of term that is inappropriate in science can be indicated with the following simple example. Suppose that we are studying a certain kind of electronic circuit. Let's suppose that we have specified two different points in circuits of this kind: an input point and an output point. And let's suppose that the following law holds:

(L1) Applications of a signal to the input cause the output signal to go to zero.

(For present purposes, we can call any nomologically necessary sentence a "law.") Given only this much information we cannot be sure that we would want to include (L1) in a statement of our theory of the circuits under investigation; we may want to subsume this fact under more general laws, for example. But we can say something about what laws would not be appropriate in our theory.

Laws that we would presumably not want to include in our theory can be constructed easily enough. Let's say that the "comprehension" of a term t that applies to events is the union of the extension of t with the set of nomologically possible events that would have been in the extension of t had they occurred. (And let the comprehensions of terms that apply to other sorts of things be defined analogously. Thus the comprehension of a term t that applies to objects is the union of the extension of t and the set of nomologically possible objects that would have been in the extension of t if they had existed, and so on.)[1] Consider a term, then, whose comprehension contains a proper subset of the comprehension of the term 'application of a signal to the input'. For example, let's call an event a

[1] I have taken the term 'comprehension' from C.I. Lewis (1946), who uses it slightly differently. Lewis says, "the comprehension of a term is the classification of all possible or consistently thinkable things to which the term would be applicable." I have restricted my "comprehensions" to the realm of the nomologically possible, i.e., to the realm of what is possible given natural laws.

conditional application of a signal to the input just in case it is an application of a signal to the input that occurs when I am standing up. The comprehension of this term is presumably a proper subset of the comprehension of the term 'application of a signal to the input', though their actual extensions might well be identical. Presumably we would not want to use this defined term in our statement of the theory of these circuits, if we assume that I am not hooked up to a switch or any other component of the circuits under investigation.

The term we have defined can be used to state various laws, though. For example, it is used in the following law:

> (L2) Conditional applications of a signal to the input cause a dropping of the output signal.

But this law is just a special case of (L1). The original law, (L1), covers all that this law covers, and more. So, if we have discovered the more general law, there is no point in our using the defined term in (L2) in any statement of our theory. That is, if we make the natural assumption that the question of whether I am standing up or not is utterly irrelevant to the operation of the circuits, then the comprehension of the defined term will be an arbitrary subset of the comprehension of the more general term, a subset that has no special significance that would make it worth mentioning anywhere in the theory. I will call terms that are inappropriate in this way "arbitrarily restricted" terms. These are the terms I want to consider.

One might think that the undesirability of these terms stems merely from the fact that they are "restricted" in the sense that more general terms can be used in their place. Since we want scientific theories to be (inter alia) as general and informative as possible, we do not want to use a restricted term if there is a more general term that can replace every occurrence in the theory of the restricted term without loss of truth value. But this is not quite right. There are cases in which we would want to use terms that were "restricted" in this sense. For example, if only one sentence of some chemical theory contained the term 'electron', the term might well turn out to be restricted just because it does not apply to things outside of the theoretical domain. We do not show the term 'electron' to be arbitrarily or inappropriately restricted in the theoretical sentence 'Electrons have negative charge', for example, just by pointing out that the term does not apply to the anode of my car battery and that if it did, the sentence would still be true. That is, we could introduce a term t' which, by stipulated definition, applies to electrons and anodes of car batteries; this new term would have a larger comprehension than the term 'electron', but it would not, presumably, be preferable. In any such case we would want to keep the restricted terms.

I do not see how to specify, in any precise and interesting way, which subset of the class of restricted terms contains all and only the restricted terms that we do not want to keep. Fortunately this is not needed for our purposes. The term 'conditional application of a signal to the input' is restricted, and we suppose that it is inappropriate in our theory because it is restricted by an arbitrary condition, a condition that has no theoretical significance. Whether I am standing up or not is presumably irrelevant to the operation of ordinary electronic circuits, so the comprehension of this restricted term is not anything we would want to pick out. Anything we would want to say about the comprehension of this term we would also want to say about the comprehension of a term that does not have the arbitrary restriction on its application. This much is clear. For the moment we will rely on this point, and leave open the question of whether the comprehension of the arbitrarily restricted term is always a proper subset of the comprehension of the preferable term that lacks the restriction, as has been the case in the examples considered so far; we will return to this issue in §6.2., below.

Putting matters roughly, we can say that, within the domain of interest, we want to use terms that apply to all and only those things that have a certain set of theoretically relevant properties. We do not want to use terms that apply only to things that have both the theoretically relevant properties and also some other properties that are irrelevant to the domain of interest, particularly when the possession of the former properties does not generally coincide with the possession of the latter. The terms that fall into this last, worst case are the ones we call "arbitrarily restricted."

3. Now we can exercise our imaginations a little and say

something more about the problems we might get into by using arbitrarily restricted terms. At the very least, the use of arbitrarily restricted terms will either complicate our theory or else restrict its scope of application. In the example discussed above, the law (L2) which contains an arbitrarily restricted term is just a special case of the more general law (L1). So a theory which contained only (L2) would be less general than a theory with (L1) unless we had some other laws which covered the same phenomena. So we have either a more complicated theory or a more restricted theory when we use arbitrarily restricted terms. This will be the case whenever arbitrarily restricted terms are used, but, as we shall see, in some cases more serious problems may arise.

Consider a causal law or other theoretical generalization that contains more than one arbitrarily restricted term. In any such case the question arises of whether the arbitrary conditions of these terms coincide, i.e., whether they hold in the same circumstances. For example, consider again the pair of terms 'application of a signal to the input' and 'conditional application of a signal to the input'. The latter term is arbitrarily restricted, and the former, we may suppose, is not. But it could happen that in studying our circuits we would discover at first only the following:

> (L3) Conditional applications of a signal to the input cause a "coincidental dropping" of the output signal,

where an event is a "coincidental dropping" of a signal just in case it is a dropping of the signal that occurs when I am standing up. (We can assume that the time between the application of an input signal and the consequent dropping of the output is so small that I never (or almost never) change position significantly in that time.) I call this a "coincidental dropping" of the signal because its arbitrary condition, that it occur when I am standing up, coincides (almost) perfectly with the arbitrary condition on the cause-term, 'conditional application'.2 Now if (L3) is in our theory, then the business of replacing the arbitrarily restricted terms becomes a bit more complicated. Simply substituting 'application of a signal to the input' for the term 'conditional application of a signal to the input' in (L3) does not preserve truth. In this case, we would want to replace both arbitrarily restricted terms at once.

We get the worst situation , though, when in stating a generalization we use terms with arbitrary conditions that do not coincide. As we have seen, if we were investigating the effects of conditional applications of a signal to the input we might discover (L3). But we would not discover even this much of the law (L1) if for some reason we restricted our investigation to a search for the relations between conditional applications of a signal to the input and "conditional droppings of the output signal to zero," where a conditional dropping of the output signal to zero is a dropping of the signal to zero which occurs

----

2. We could make the coincidence of the conditions really perfect by defining a coincidental dropping of the signal to be a dropping of the signal that is caused by an input that is applied when I am standing up, but this would make the example less natural and, I think, less like the actual cases we might encounter.

when I am sitting down; the arbitrary conditions on the correct applications of these terms do not coincide in any situation. So restricting our attention to the extensions of arbitrarily restricted terms will not only restrict the scope of our theory, but, as in this case, it may also more or less completely conceal just the sort of relations that we would like to discover. The degree of concealment will depend on the degree of coincidence of the respective arbitrary conditions.

In summary then, we have presented a defense of the following methodological principle:

> (P1) Arbitrarily restricted terms should be avoided in the statement of theories; when unrestricted replacements for these terms are not available, they should be sought.

As we have pointed out above, the "seeking" for the appropriate replacements amounts to the seeking of the theoretically relevant properties in terms of which the replacement may be defined. And as we have seen, the severity of the adverse consequences of using arbitrarily restricted terms depends on features of each particular case. In cases where the adverse concequences are minimal, as, for example, when the arbitrary conditions on the application of a term hold in all possible situations, the injunction to seek a more appropriate replacement is, of course, less pressing.

4. Now let's consider ordinary, pre-theoretical mental terms like 'sees a robin' and 'sees a dove emerging from the magician's pocket'. Perhaps the truth of Ryle's suggestion about such terms is that they are analogous to terms like 'conditional application of a signal to the input'. If we can show that these ordinary mental terms have arbitrary conditions on their application that would render them arbitrarily restricted in psychology, then we can conclude that they should be avoided in psychological theory for the reasons just discussed. Many philosophers and psychologists have argued that mental terms are inappropriate for other reasons as well, but we do not need to consider here whether any of these other claims are correct. We do not even need to assume that any ordinary mental terms are ever actually used in the statement of a psychological theory (which, of course, they are) in order to argue that if they were, they would be arbitrarily restricted by some conditions on their application.

Ryle has pointed out, as others had before him, that certain ordinary mental terms have "contextual" conditions on their application. In the passage quoted above, for example, he considered some spectators witnessing doves emerging from a conjuror's pocket and said, "They could not have seen it happen unless it happened, and unless it happened in such a place and at such a distance and in such a light that it was visible to them and unless their eyes were open, properly directed and focused and so on." Let's suppose that this claim is correct, that such conditions must indeed be satisfied if the spectators can be correctly said to have seen the event. On our construal, Ryle's point is that appropriate scientific terms do not have such conditions on their application.

Our problem, then, is to decide whether these conditions are "arbitrary" conditions on the application of psychological terms, whether they are suitably irrelevant to what psychology is about. And before this problem can be properly dealt with, we need some assumptions about the domain of psychology. Unfortunately, it is not very clear what psychology is (or should be) about, and a full consideration of this controversial matter is beyond the scope of this chapter. Since our interest is primarily to be clear about how certain methodological principles apply in any domain, we will just make some assumptions here about the domain of interest and keep in mind that analogous methodological principles will apply in other domains. It will be convenient to assume, then, that our interest in psychology is to predict and explain behavior, where we take the term "behavior" very loosely. We will assume that virtually any kind of physiological state or event or physical movement that is caused by an event in the nervous system may be something we want psychology to explain. We will return later to the question of how our conclusions depend on this assumption.

Given this view of the domain of psychology, it is natural to hold that the contextual conditions on the correct application of the predicate 'sees the event' or of the term 'seeing the event' that are mentioned by Ryle are arbitrary in psychology. Consider, for example, whether doves really emerged from the magician's pocket. Is this question relevant to the explanation of the subject's behavior? We naturally assume that it is not; the subject will act in the same way whether he really saw the doves or just mistakenly believes that he saw them. Presumably the subject's behavior can be explained on the basis of his sensory inputs and his internal states, if it can be explained at all; the question of what actually caused the sensory input to be such as it was is thus irrelevant. This is, I suppose, an empirical assumption, though it is eminently plausible. It is conceivable that some part of a subject's behavior is contingent upon the actual occurrence of such an event, just as it is conceivable that the operation of some ordinary electronic circuit depends upon whether I am standing up or not. It is conceivable but not believable. We can call such assumptions about the factors relevant to the domain of interest, "assumptions of causal closure".

We will not undertake to catalog all the ordinary mental terms that are arbitrarily restricted in psychology. Many mental terms that contain some form of a factive verb clearly have arbitrary conditions on their application which would restrict their application in psychology. And less obvious arbitrary conditions on certain mental terms are apparently suggested by causal theories in the philosophy of mind, by some theories about the entailments of sentences containing mental terms that include proper names, indexicals, or natural kind terms, and by theories about what Burge(1979) has called "socially dependent features of cognitive phenomena."[3]

In any case, it is clear that when our empirical assumptions about the causes of behavior are taken together with our methodological claim that arbitrarily restricted terms should be avoided in science, we have some substantial guidelines for psychology. To take one that has attracted some interest recently, we have a principle that is very much like what Putnam(1975) has called "methodological solipsism". He says,

> ...an assumption which we may call the assumption of methodological solipsism...is that no psychological state, properly so called, presupposes the existence of any individual other than the subject to whom that state is ascribed.(p.220)

Putnam considers whether psychology should adopt this assumption, and he distinguishes the assumption from a view that is more common in philosophy, viz., "that no psychological state presupposes the existence of the subject's body even." So when this passage is taken in its original context, it appears to be suggesting an assumption that is similar to a methodological principle for which we have provided a defense, namely,

(MS) Avoid using terms that have as a condition on their

[3] Factives are discussed by, e.g., Kiparsky and Kiparsky(1970). Causal theories in the philosophy of mind are discussed by, e.g., Grice(1961), Martin and Deutscher(1966), Goldman(1967), and Wilson(1972). Putnam(1975, ch. 12), for example, discusses mental terms that contain indexicals and natural kind terms. And the social and historical context presupposed by certain sentences containing mental terms is emphasized by, e.g., Wittgenstein(1958), Ryle(1954) and Burge(1979). A complete bibliography of the relevant literature would be very extensive.

application the existence of any person other than the subject to whom the term is applied.

So, for example, the predicate 'sees Jimmy Carter' should be avoided in psychology, since it has as a condition on its application the existence of Jimmy Carter. We want to reject any such term in favor of a term that would apply regardless of the existence of any other particular individual. Of course, the view we have defended supports not only the principle (MS) but also analogous principles that deal with terms that presuppose the existence of robins, cabbages, rocks, and so on.

Although we do not really have any very good idea of what will be needed to explain visual perception (if it can be explained at all), it is surely implausible that there are lawlike relations between the actual presence of other individuals and visual perception or anything else that falls within the psychological domain we have indicated. Once again, it should be emphasized that we are trying to stick to cases that seem clear; even these are of some interest.

It also seems that there are clear cases to establish that the rather vaguely apprehended distinction between the "contextual" conditions and the "internal" conditions of an organism does not, given our natural assumptions of causal closure, coincide with the distinction between conditions that are arbitrary in psychology and those that are not. It is easy to think of "internal", "non-contextual" conditions that are arbitrary in psychology. Consider, for example, the predicate 'knows that his own blood is AB positive'. No one can know that his own blood is AB positive unless his blood is AB positive, but this "non-contextual" condition certainly looks quite arbitrary in psychology. Or consider the term, 'feels his ulcer acting up'; no one can feel his ulcer acting up unless he really does have an ulcer. So there appear to be many common mental terms that have conditions on their application that are internal but arbitrary nevertheless. And, on the other hand, there presumably are contextual conditions that are not arbitrary in psychology. Certainly the behavior of an organism is contingent on the environment's meeting some conditions! This much is clear despite the difficulty of finding laws that govern these organism-environment relations. So there are internal conditions that are arbitrary, and there must be contextual conditions that are not arbitrary. But the precise limits of the arbitrary is a matter for future psychologists to discover.

5. Some psychologists would certainly want to object to our methodological proposals. One way of undermining them would be to refute our arguments for our "assumptions of causal closure." Let's consider how one such objection might go. A psychologist might object to our comparison of the supposition that a subject's behavior depends on the actual presence of other individuals with the supposition that the operation of ordinary electric circuits depends on whether I am standing up. Certainly there are differences between these two cases that are relevant

to the present discussion. We would assume that my being standing up was an arbitrary condition in the circuit theory even if there happened to be a correlation between the obtaining of the condition and, say, the production of a certain output; this correlation could presumably be broken in an experimental situation, and this would support our claim about the arbitrariness of the condition. But in psychology, as in the special sciences generally, we are typically not concerned with determinate, exceptionless laws of the sort we have in circuit theory, and so the business of deciding what conditions are arbitrary in psychology becomes considerably more complicated. If the psychological laws are probabilistic, then we must judge the relevance of conditions on the application of their terms accordingly. But, in fact, the situation is even more difficult than this.

Many of the generalizations that one finds in psychology not only allow exceptions, but also do not make any precise, testable claim about the improbability of the exceptions. Grice(1975) has suggested that some or all of the laws containing ordinary mental terms are of this kind. And Fodor(1975, pp.13-25) has suggested that it should be expected that the laws of the special sciences generally will have exceptions. So given the present state of generalizations in psychology, decisions about what conditions are arbitrary may become a bit sticky, as does the methodology of theory confirmation generally.

Perhaps this is the point that a psychologist would try to exploit in an objection to our assumptions of causal closure. Consider some predicate like 'sees a rigid object'. Couldn't one grant that this predicate does not apply to a subject unless (*inter alia*) there is a rigid object in his field of vision, and that the effect of this object on the subject could have been just the same regardless of whether there was really a rigid object there or only, say, a hologram of some rigid object, and nevertheless deny that this condition is arbitrary? Suppose one held, for example, that we can do fruitful work finding psychological generalizations that hold only in certain normal human "ecological" situations.[4] Then our argument for the arbitrariness of certain contextual conditions has no force; it merely points up the acknowledged fact that these generalizations will break down or fail to apply outside of the "normal" contexts.

----

[4] I think there are grounds for construing the position of some Gibsonian psychologists this way. (Cf., e.g., Gibson, 1966, and Mace, 1974.) Certainly this construal misses whatever virtues there are in taking ecological considerations seriously in perceptual psychology, e.g., to suggest what stimuli the senses are likely to be responsive to. But Gibson apparently would object to our position. For example, he says (in his usual inscrutable style), "The classical theories of perception...explain both perception and misperception, both detection and illusion, with the same assumptions... There is a lack of logic here. If misperception is the opposite of perception, the law of association or the law of sensory organization cannot apply to both at the same time. The same principle should not be used to explain why perceiving is so often correct and why it is so often incorrect. A theory of perception should certainly allow for misperception, but it can hardly at the same time be a theory of misperception. (1966, p. 287) I cannot make out any sound argument here.

But this will not do. Suppose that in an ecologically "normal" situation, seeing a rigid body has a certain effect on, say, behavior and internal state. Then, as we have urged, seeing something that looks exactly like that rigid body would have had the same effect, and so we should look for a term that would apply in both cases, i.e., a term that is not restricted by the condition that a rigid body actually be present. A theory with this unrestricted term would constitute an <u>improved</u> extension of the "ecological" theory. I cannot imagine any good reason for thinking that making improvements of this sort whenever we can would impede the project in any way. Let the ecological approach produce its theory (if it can), and we will produce an extension of it that has wider application. No grounds have been offerred here for thinking that such a "naturalistic psychology" must fail. Rather, we have argued that if such a project succeeds, so will a better one.

6.1. There may be a temptation here to say that what psychology can really explain is just "part of" an event like seeing Carter, or "part of" whatever other mental event, state or process falls in the extension of an arbitrarily restricted term. But whether this sort of metaphysical claim is intelligible and correct or not, it is nevertheless of interest to consider what events are in the extension of the appropriate replacement for an arbitrarily restricted term. What predicate would be an appropriate replacement for the predicate 'sees Carter', for example? What event occurs when one sees Carter that we <u>should</u>

expect the theory to explain? One proposal is offerred in the following passage from Fodor(1975):

> ...one might, as it were, 'construct' a nonrelational propositional attitude corresponding to each relational one by 'dropping' such conditons on the ascription of the latter as constrain nonpsychological states, events or processes. So, to a first approximation, 'rationally believing' corresponds to 'knowing' in the sense that an organism believes that a is F iff the organism satisfies all the conditions on knowing that a is F except the factivity condition. In a similar spirit, 'seeming to see' corresponds to seeing, 'seeming to hear' corresponds to hearing, etc. (p.76n)

It does not seem to me that any of these suggestions provides quite what we need. For example, it is not at all clear that 'seems to see Carter' would be the appropriate replacement for the arbitrarily restricted 'sees Carter'.

One might take instances of the schema 'S seems to see x' as equivalent to the corresponding instances of 'It seems to S that he sees x', but this is surely not the reading Fodor has in mind. A subject might actually <u>see</u> a robin without its seeming to <u>him</u> that he sees a robin; he might not have any idea about what sort of animal he saw. Presumably Fodor intends us to read instances of 'S seems to see x' as equivalent to something like the corresponding instance of 'S is having a visual experience as of x'. The problem with this proposal is that we don't really have any clear idea of what having the required visual experience is. For example, could one have had "a visual experience as of Carter" even if Carter had never existed? We need to assume so, if the term 'has a visual experience as of Carter' is not to be restricted by the condition that Carter exist. But if one could have had this visual experience even if Carter had never existed, then one might begin to wonder what sort of experience it is. How could we individuate this experience?

The point I want to make is really just the obvious one. Suppose that we actually have psychological laws that contain some predicate like 'sees a robin'. (It is not really likely that we would have a theory with laws about perceiving robins or perceiving Carter, but our examples serve to illustrate some general points.) Presumably the behavior (and internal states) of a subject depends on his sensory inputs and his internal states; as we noted above, what actually causes the input to be such as it is is presumably irrelevant. We want a term that will apply to any event which is like seeing a robin in psychologically relevant respects. The point is that we don't know a priori what the psychologically relevant respects are. It is not at all clear that the mere fact that the subject's visual experience was like seeing a robin in whatever respects are psychologically relevant guarantees that it would be correct to say he seemed to see a robin.

I do not see any good grounds for supposing that merely choosing related terms that are not relational will generally give us appropriate replacements for arbitrarily restricted terms. It might well be that English does not contain the terms we will want to use in our theory. In such a case we could resort to neologism, or there is the option suggested by Ryle in the passages quoted at the beginning of this paper, viz., that ordinary mental terms may be used in psychology with non-standard meanings. This is what we might expect to find most often in mentalistic psychology, since the comprehensions of the appropriate unrestricted terms will often come to be recognized only gradually, in the course of developing the theory.

6.2. There is another interesting issue with regard to finding appropriate replacements for arbitrarily restricted terms, namely, the question of whether the comprehension of an arbitrarily restricted term will always be a subset of the comprehension of an appropriate replacement for that term. I will not attempt to decide the question; I will just point out one example in which one might plausibly argue the the comprehension of the replacement will not include the comprehension of the arbitrarily restricted term.

Consider a theory with terms like 'seeing a robin' which apply to events, as opposed to, or in addition to terms like 'sees a robin' which apply to people and other sorts of things. If we grant that 'seeing a robin' truly applies to an event only if a robin is present, and that the presence of a robin is an arbitrary condition in psychology, then, for our theoretical pursuits, we ought to look for a term that is not restricted by this condition. The interesting point is that in this case one might hold that the appropriate replacement is one whose comprehension does not include the comprehension of 'seeing a robin' as a proper part. A term whose comprehension does include the restricted comprehension as a proper part would presumably be preferable to the original term, but it seems that in a case like this we might want to use a term whose comprehension is disjoint from the comprehension of the original term. Let's consider why this might be so.

Suppose that we are looking for a replacement for the term 'seeing a robin', a replacement whose application is not restricted by the condition that a robin be present. One might suppose, as a first guess, that the psychologically relevant event that occurs when I see a robin is the having of a certain sort of visual experience, or _something_ like that, as Fodor apparently means to suggest in the passage discussed above. So we might guess that the appropriate replacement for 'seeing a robin' would be a term t that applies to a subject's seeing a robin and to a subject's having certain visual experience, e. Presumably everything we would want to say about the comprehension of 'seeing a robin' we would also want to say about the comprehension of t. But if t will serve our purpose here, then it is plausible to suppose that the term 'having visual experience e' would serve the purpose more naturally, even though it is plausible that the comprehension of this term is disjoint from the comprehension of 'seeing a robin'. The reason for thinking that the comprehensions of these terms are disjoint is that it is plausible that each event in the extension of 'seeing a robin' has the presence of a robin as an essential property; it is plausible that my seeing a robin is an event which could not have occurred if there were no robins around. And, on the other hand, one might suppose that each event in the comprehension of the term 'having visual experience e' would surely be an event that could have occurred regardless of whether a robin were present or not. If this is correct, then the two comprehensions must be disjoint. We know that the latter comprehension is larger though, since whenever a subject sees a robin he has the psychologically relevant visual experience e, and one may sometimes have this visual experience without seeing a robin. So if everything we want to say about seeing a robin we also want to say about having visual experience e, then the latter claim is the one we want to make even if the comprehensions of the respective terms are disjoint.

Accepting this conclusion would bring our view a little closer to Ryle's. We could urge that an "ideal" psychology, i.e. a psychology that did not contain any arbitrarily restricted terms, would probably not mention any events like seeing or hearing. But it should be noted that many philosophers would object to the account of seeing that was used to defend this conclusion. Some would want to argue that (i) my seeing a robin is an event that could have occurred even if a robin had not been present, or (ii) that the idea that events have essential properties is somehow confused. Both of these alternatives have their advocates. The former view is presupposed by certain causal and functionalist theories of mental states and events. Certain causal theories suggest that my perceiving x _is_ my having a certain sense-impression, a sense-impression caused by x. (Cf., e.g., Grice, 1961.) If I could have had that sense-impression even if x had not existed, then, on this view, my perceiving x is an event that could have occurred even if x had not existed. Similar views are suggested by the following passages,

> Though it is always logically possible, for any given effect, that it exist without being caused, we would not say that it would, in that case, continue to be an effect. Similarly, we might grant, if the emotion/object relation is merely a causal one, then either of its terms might exist apart from that relation. But we needn't grant that either of the terms would, in that case, continue to be the same sort of emotion, or even to be any sort of emotion at all. (Aquila, 1974, p.280)

> Geach thinks that descriptions which specify thoughts by reference to certain of their contextual relations (perhaps by reference to their distal causes) are the ones which pick out their essential properties; whereas, I think it's descriptions which specify functional properties that do so. (Fodor, 1980, p.102)

If these views are correct, then the argument presented for the view that 'seeing a robin' and 'having visual experience e' have disjoint comprehensions is unsound. This would not exclude the possibility of other examples, though, in which we would want to allow that the comprehensions of appropriate replacements for arbitrarily restricted terms do not include the comprehensions of the terms they replace.

7.1. Now that we have proposed some methodological guidelines for psychology on the basis of some empirical assumptions, it is of interest to consider their impact on psychology. First of all, it should be noted that there seems to be a tendency to assume that the penalties of using arbitrarily restricted terms are more serious than has been claimed here. For example, it has been suggested that Ryle's arguments show that "strictly speaking, there can't be a psychology of perception." (Fodor, 1980, p.64) And one hears claims like the following,

> In satisfying...demands for comprehensive and exact explanation, I shall find that I am investigating the mechanisms of the performance rather than the performance identified and described as a social and cultural phenomenon, and as an expression of thought. The descriptions of behavior which can be fitted into a scheme of scientific explanation must be appropriately determinate and exact; they must not depend for their interpretation on the context of use...Psychological terms, drawn from the common vocabulary, do not satisfy these...conditions...The truly scientific study of human nature, necessarily concerned with universal laws, will leave the explanation of human thought, conscious and unconscious, untouched, except for the abstractions of logic... which are detached...from any particular social context. (Hampshire, 1978, pp.66-67, 68)

But as we have seen, the mere fact that a term is arbitrarily restricted by contextual conditions does not show that it will not occur in any laws. This point was demonstrated by our laws (L2) and (L3) which contain the arbitrarily restricted terms 'conditional application' and 'coincidental dropping of a signal'. To get generalizations as reliable as these in perceptual psychology would be significant and exciting. And, as we have seen, appropriate replacements for arbitrarily restricted terms will often apply to everything the restricted terms apply to, and more. The mere fact that a term is arbitrarily restricted does not show that we cannot give any scientific account of the things in its comprehension.

What has been argued here is only that if there are laws containing arbitrarily restricted terms, we can expect to find laws of broader application that are preferable, and that the search for laws containing arbitrarily restricted terms may divert our attention from the more general relations that we would like to discover. (See §3.) Of course, it could turn out that some or all ordinary mental terms have problems more serious than just that they are arbitrarily restricted in psychology, but here we are only concerned with the penalties of using terms that are arbitrarily restricted.

Let's consider a little more carefully what penalties we would expect psychology to be paying for its use of arbitrarily restricted mental terms. The amount that arbitrarily restricted terms will restrict the application of the laws in which they occur depends on how restrictive their arbitrary conditions are. And it was argued above that restricting our attention to relations between extensions of arbitrarily restricted terms may conceal relations that we would like to discover, and that the degree of concealment depends on the degree to which the arbitrary conditions on the application of the relevant terms coincide. If they coincide perfectly, then we would expect only restricted application. So these are two factors that should be kept in mind when we look at the use of arbitrarily restricted terms in psychology: how restrictive are the conditions on the applications of the terms, and do the conditions on the terms of interest in a certain area coincide?

Psychologists apparently do make claims about ordinary mental states and events. They propose theories about concept learning, perception, natural language understanding, etc. And these theories include generalizations that contain ordinary mental terms. It could be that in psychology these terms are used with special technical senses, but, if they are, it is not widely recognized that they are. So such terms are at least interesting candidates for consideration. So we want to ask: Do these theories fail to respect our principles? And if they do, do these theories suffer the ill-consequences that we expect to attend the use of arbitrarily restricted terms, and are these restricted terms easily eliminated in favor of more general terms? It should be kept in mind here that our methodological principles apply to the generalizations that would occur in the statement of a theory, whereas a good deal of scientific literature discusses particular applications of theoretical generalizations which are themselves sometimes only implicitly suggested.

7.2. The psychological claims that would be of the greatest interest to us are those that contain terms which we would expect to be very inappropriate. One example that springs to mind is the predicate 'knows' and its cognates. Many terms containing some form of the verb 'to know' have conditions on their application that would be arbitrary and restrictive in psychology. Predicates of the form "knows that p", for example, apply to a subject only if the predicate complement is true, and the condition that it be true will often be arbitrary in psychology. So one might expect that psychologists would not work on theories about knowledge, but this is apparently not the case. Many psychological claims about knowledge can be found even in the most recent literature. Cognitive psychologists have shown great interest, for example, in claims about how knowledge is mentally represented, organized and used, and in claims about knowing a natural language. Let's consider briefly an example from each of these areas of interest.

One paper that has been very influential in recent work on how human knowledge is represented is Minsky's(1974) paper on "frames". This paper proposes (<u>inter</u> <u>alia</u>),

> (E1) Human knowledge is represented in a system of interconnected structures called "frames".

One might argue (roughly) that the term 'human knowledge' is arbitrarily restricted since it does not apply to false or inaccurate beliefs which are also, or could be, represented in the frame system. That is, the internal, mental representation of information in frames is surely not contingent on the truth of that information. But it is clear that the holders of (E1) would want to hold that inaccurate information is also represented in the frame system, so we can conclude either that Minsky is using the term 'human knowledge' much more loosely than it is ordinarily used, or that he intends to make a more general claim than (E1) that he has not stated explicitly. On this latter assumption, then, (E1) does contain an arbitrarily restricted term, but this is easily recognized and a more general claim could be made. A well-developed theory would indicate clearly what that more general claim would be. As a matter of fact, the proponents of this framework apparently want to claim that frames are used to represent not only the content of beliefs (regardless of their truth), but also the content of propositional attitudes generally, the information processed in visual perception, etc. So it looks like a new term will have to be coined in order to make the most general claim about what is represented in frames.

Psychologists, linguists and others have also expressed great interest in theokries about knowledge of language. Chomsky, for example, has argued that,

> (E2) If a person knows a natural language L, then that person has internalized a grammar of L. (See, e.g., Chomsky, 1975, p.304.)

One might argue that the predicate 'knows a natural language L' is arbitrarily restricted because no one can be in the extension of 'knows a natural language L' (for any L) unless L really is a natural language, and this condition is arbitrary in psychology. If there were no such language, but a subject's sensory input were just as if there were, then the subject would presumably internalize a grammar anyway; the actual use of the language by other people is really irrelevant to the psychological account of the matter.

Once again, we cannot go into this matter in detail, but I suspect that Chomsky and other advocates of this view would reject our argument on the grounds that it is a mistake to presume that a natural language must be one that is or has been used by a community of speakers. If a subject recieved sensory input that caused him to internalize some grammar of a language, that language is ipso facto a natural language, a possible human language, regardless of whether it is actually used by a community of speakers. If this is the right account, then our argument that 'knows a natural language L' is arbitrarily restricted fails. So we have no good reason to think that this knowledge claim is inappropriate in this way.

Another set of terms that are prime suspects for being arbitrarily restricted are the terms used in theories of perception. So let's briefly consider an example from this domain. One area of perceptual psychology that has attracted the interest of many psychologists is the perception of natural language. One claim that has been accepted by many recent theorists is the following:

> (E3) Recognizing an acoustic token t of a spoken sentence involves formulating a syntactic representation of that sentence token. (See, e.g., Marslen-Wilson and Tyler, 1980, and references cited there for a discussion of this claim.)

This claim contains the term 'recognizing an acoustic token t of a spoken sentence' which seems to be arbitrarily restricted, since no one can recognize a spoken sentence token t unless there actually is such a token, and this condition is presumably arbitrary in psychology. If a subject's sensory input were as if such a token had been produced in his presence, he would presumably (according to adherents of this theory) produce a syntactic representation of the sentence whether it was actually spoken or not. This point would probably be conceded by anyone who holds (E3). But notice that this condition is not very restrictive; fortunately, a subject who receives sensory input just as if he had heard a spoken sentence has, as a matter of fact, almost always heard a spoken sentence. And, although theorists would probably recognize the restricted nature of their claim, it is not entirely clear how to formulate a better one; it is not obvious what term should be used in place of this arbitrarily restricted term.

It is interesting to note that some recent theories that are concerned with perception do not use arbitrarily restricted perceptual terms at all. We have, for example, theories of "visual information processing" instead of theories of "visual perception". These theories typically take special care to account for cases of misperception, i.e., for visual illusions. And we find claims like the following:

(E4) The difference between relative positions of visual images on the retina of each eye is measured and used to compute an estimate of depth; this involves (inter alia) producing descriptions of the images that are rather like sketches. (See, e.g., Marr, 1979, and references cited there.)

There are apparently no ordinary perceptual terms in this claim, though it is of course a claim that would be considered relevant to the psychological explanation of actual cases of perception.

7.3. In sum, these examples are perhaps typical of what we would expect to find in psychology. We would not expect to find severely restricted terms. Neither would we expect to find terms in psychology whose conditions are highly non-coincidental, since we would expect to find arbitrary conditions that hold most of the time. We do find claims about knowledge that are, perhaps, restricted. But it is no surprise that we find predicates like 'knows a human language' rather than predicates like 'knows Carter is present'. We can explain why these sharply restricted terms are avoided, and we can do so without committing ourselves to the implausible view that there can be no scientific generalizations whatever about the events in the extensions of arbitrarily restricted terms.

8. With regard to the assumption we made about a domain for psychology, there are two points to be made. In the first place, we have not defended the view that work in this area is likely to

be fruitful. Indeed, considering the amount of effort that has been invested in this program, there has been disappointingly little progress in discovering any laws underlying any interesting aspect of the (supposed) dependence of behavior on sensory stimulation and internal state. It may be that this domain will never prove to be fruitful; it may be an area in which we will be unable to find significant psychological laws.

The second point to be made about our choice of a psychological domain has been noted already, that our conclusions about what terms are arbitrarily restricted in a theory depend crucially on the theoretical domain, and so they will not generally hold in other domains. We argued, for example, that the presence of other individuals is an arbitrary condition in any theory that aims to explain individual behavior on the basis of sensory stimulation and internal state. But surely there are other domains in which this condition is not arbitrary, such as domains in which the individual is viewed in a certain context. Consider, for example, the domains of human ethology, evolutionary biology, and sociology. And presumably there are also "psychological" domains in which it is not at all obvious that our contextual conditions would be arbitrary, as perhaps in certain theories in psychophysics or social psychology. There may even be other psychological domains of this kind that we have not distinguished yet, domains in which we could do fruitful scientific work.

Perhaps these considerations about the fruitfulness of various domains of inquiry are what we need to shed some light on Putnam's comments on methodological solipsism. He says,

> Making this assumption is, of course, adopting a restrictive program...(We shall...refer to mental states that are permitted by methodological solipsism as 'psychological states in the narrow sense'.) Only if we assume that psychological states in the narrow sense have a significant degree of causal closure (so that restricting ourselves to psychological states in the narrow sense will facilitate the statement of psychological laws) is there any point...in making the assumption of methodological solipsism. But three centuries of failure of mentalistic psychology is tremendous evidence against this procedure, in my opinion. (1975, pp. 220-221)

This passage suggests that Putnam anticipated the sort of defense of methodological solipsism that was developed here, and that he thinks that since mentalistic psychology has failed, we should look for other more fruitful psychological domains in which there might well be no argument for accepting methodological solipsism. But if he did anticipate our argument, then his remarks are surprising. In the first place, our defense of methodological solipsism does not justify the adoption of any restrictive practices. If we had defended the view that the prospects of finding significant generalizations containing arbitrarily restricted terms were hopeless, then we might want to adopt the "restrictive" program of avoiding them entirely. But this strong conclusion is not defended here. We allow that significant generalizations containing arbitrarily restricted terms may be forthcoming, and recommend only that these terms are not the most general that we could use and so we will want to seek the more general terms. The second peculiarity of Putnam's remarks is that he is apparently suggesting that the assumption of methodological solipsism should be rejected because mentalistic psychology has failed. This is a little odd, though, since in the first place, mentalistic psychology has not respected this assumption; and in the second place, the mentalistic psychology we have developed hardly begins to to exhaust the field of possible theories that do respect this assumption. And the third peculiarity of Putnam's remarks is that he seems to be suggesting that psychologists should reject methodological solipsism despite the sort of argument that has been presented here. Perhaps, then, he is suggesting that methodological solipsism is restrictive and to be discarded even in theories that aim to explain individual behavior, theories in the domain we have been considering. But to reject our argument and maintain this view, one would need to deny either the plausible principles of causal closure discussed above or else the general methodological principles on which our defense of methodological solipsism rests. Neither of these options looks attractive. So, in any case, it is not clear what Putnam means to suggest.

9. Apart from the related questions about what psychology ought to <u>explain</u>, the views that have been proposed here are, I think, the most significant that we can derive from the suggestion of Ryle's with which we began. I suspect that Ryle and some other philosophers have thought that more devastating conclusions would follow, but our conclusions are really fairly strong. We have advanced some general methodological principles

that are supposed to apply to ahy possiblə science of the relevant domain, so the mere fact that they do not conflict with present day psychology (if indeed they do not) affords them but little merit. Some philosophers and psychologists would certainly want to argue that our conclusions are too strong, but it is at least not obvious how such a challenge could be carried through successfully.

BIBLIOGRAPHY FOR CHAPTER II

Aquila, R.E. (1974) Emotions, objects and causal relations. *Philosophical Studies*, *26*, pp279-285.

Burge, T. (1979) Individualism and the mental. In P.A. French and T.E. Uehling, Jr., *Midwest Studies in Philosophy, 4, Studies in Metaphysics*. Minneapolis: University of Minnesota Press.

Chomsky, N. (1975) Knowledge of language. In K. Gunderson, ed., *Language, Mind, and Knowledge*. Minneapolis: University of Minnesota Press.

Davidson, D. (1970) The individuation of events. In N. Rescher, ed., *Essays in Honor of Carl G. Hempel*. Dordrecht, Holland: D. Riedel.

Davidson, D. Psychology as philosophy. In S.C. Brown, ed., *Philosophy of Psychology*. New York: Barnes and Noble.

Fodor, J.A. (1968) *Psychological Explanation: An Introduction to the Philosophy of Psychology*. New York: Random House.

Fodor, J.A. (1975) *The Language of Thought*. New York: Crowell.

Fodor, J.A. (1980) Methodological solipsism considered as a research strategy in cognitive psychology. *Behavioral and Brain Sciences*, *3*, pp.63-109.

Gibson, J.J. (1966) *The Senses Considered as Perceptual Systems*. Boston: Houghton Mifflin.

Gibson, J.J. (1979) *The Ecological Approach to Visual Perception*. Boston: Houghton-Mifflin.

Goldman, A. A causal theory of knowing. *Journal of Philosophy.*, *64*.

Grice, H.P. (1961) The causal theory of perception. Aristotelian Supplementary Volume, 35, pp.121-168.

Grice, H.P. (1975) Method in philosophical psychology. Proceedings and Addresses of the American Philosophical Association, 48.

Hampshire, S. (1978) The illusion of sociobiology. The New York Review of Books, 25(15), pp.64-69.

Kiparsky, P. and Kiparsky, C. (1970) Fact. Reprinted in D.D. Steinberg and L.A. Jakobovits, eds., Semantics: An Interdisciplinary Reader in Philosophy, Linguistics and Psychology. New York: Cambridge University Press, 1971.

Kripke, S.A. (1972) Naming and necessity. In D. Davidson and G. Harman, eds., Semantics of Natural Language. Boston: D. Riedel.

Lewis, C.I. (1946) The modes of meaning. Reprinted in J.F. Rosenberg and C. Travis, eds., Readings in the Philosophy of Language, Englewood Cliffs, New Jersey: Prentice-Hall, 1971.

Mace, W.M. (1974) Ecologically stimulating cognitive psychology: Gibsonian Perspectives. In W.B. Weimer and D.S. Palermo, eds., Cognition and the Symbolic Processes. Hillsdale, New Jersey: Lawrence Erlbaum.

Marr, D. (1979) Representing and computing visual information. In P.H. Winston and R.H. Brown, eds., Artificial Intelligence: An MIT Perspective, Volume II: Understanding Vision, Manipulation, Computer Design, Symbol Manipulation. Cambridge, Massachusetts: MIT Press.

Marslen-Wilson, W. and Tyler, L.K. (1980) The temporal structure of spoken language understanding. Cognition, 8.

Martin, C.B. and Deutscher, M. (1966) Remembering. Philosophical Review, 75, pp.161-196.

Minsky, M. (1974) A framework for representing knowledge. Reprinted in P.H. Winston, ed., The Psychology of Computer Vision. New York: McGraw-Hill, 1975.

Putnam, H. (1960) Minds and machines. In S. Hook, ed., *Dimensions of Mind*. New York:MacMillan.

Putnam, H. (1975) *Mind, Language, and Reality: Philosophical Papers, Volume 2*. Cambridge:Cambridge University Press.

Ryle, G. (1949) *The Concept of Mind*. New York: Barnes and Noble.

Ryle, G. (1954) *Dilemmas*. Cambridge: Cambridge University Press.

Thomson, J.J. (1977) *Acts and Other Events*. Ithaca, New York: Cornell University Press.

Vendler, Z. (1957) Verbs and times. Revised and reprinted in Z. Vendler, *Linguistics in Philosophy*. Ithaca, New York: Cornell University Press, 1967.

Wilson, J.R.S. (1972) *Emotion and Object*. New York: Cambridge University Press.

Wittgenstein, L. (1958) *Philosophical Investigations*. Translated by G.E.M. Anscombe. New York: MacMillan.

CHAPTER III

A THEORY OF COMPUTING SYSTEMS

TABLE OF CONTENTS

CHAPTER III

A THEORY OF COMPUTING SYSTEMS

1. Introduction

The development of computing machines with remarkable abilities is certainly due, in large part, to the fact that we are able to describe the complex internal operations of these machines as "information processing." The possibility of finding similar computational accounts of the operation of naturally occurring systems like the nervous systems of humans has aroused considerable excitement among psychologists and neurophysiologists. The development and assessment of any computational account of the nervous system, however, requires some general understanding of what it is for a physical system to be a computing system of a particular kind. Computer scientists have provided theories that apply to our electronic computers, but psychologists need theories that have clear application to physical systems that are really quite different, and they need theories that answer questions that are not of general interest to computer scientists and engineers. Psychologists have made claims about what sort of computing language the human nervous system uses, for example, but the content and credibility of such claims are obscure until we have a general understanding of what it is for a physical system to use a computing language. In this chapter an elementary theory of computing systems will be presented which will have clear application to such issues in psychology. This theory of computation will be applied to some psychological issues in the next chapter.

The hardest question that our theory of computation should go some way towards answering is the one just suggested: Given any particular programming language, L, what is it for a physical system to use L? Answering this question requires that we say something about what it is for a physical system to execute the programs of the language, and this, in turn, requires that we explain what it is for a physical system to compute a function. A program may be used simply to specify the function computed by a system, or simply to specify the program computed by the system, or to specify the program that is represented and in control of the computational processes. Other accounts of the role of programs in explaining the behavior of physical systems have failed to distinguish these three roles, yet, as we shall see, these distinctions are crucial to understanding claims about what programs are used and claims about what programming language is being used.[1] As we will see, defining what function is computed by a program and defining what it is for a system to compute a program depends on what sorts of statements or

[1] Other accounts of the role of programs in explaining the behavior of physical systems are provided in Fodor, 1968, ch.4; Newell and Simon, 1976; Cummins, 1977; Haugeland, 1978; and Pylyshyn, 1980.

instructions our programs are allowed to include. But programming languages contain so many sorts of statements, each of which requires special treatment, that our task quickly becomes overwhelming, as we will point out below. So instead of providing a fully general and precise account that applies to any programming language, we will provide a relatively precise account for only one simple programming language. The way in which the account should be extended to cover other languages and the difficulties involved in such extensions will be rather vaguely indicated.

We have chosen for detailed consideration a language suggested by Scott(1967). It is a simple language capable of expressing any program that can be represented by a flowchart. We want to define the language in a way that does not depend on any particular implementation on a machine, since we want our account to be as general as possible. In particular, we want our account to cover physical systems other than the artefacts produced by our electronics industry. The definition suggested by Scott serves admirably in this respect, so only minor changes have been made in it. The theory of computation that has been developed by Scott and others has been presented in introductory texts by Clark and Cowell(1976) and Stoy(1977), but we make use of only the most very basic and uncontroversial part of the approach.

2. A simple computer

The basic elements of our account of computing systems can be illustrated with a simple example. Consider what would suffice to make some physical system P̲ an adding machine. We would need to be able to interpret some states of the system numerically. Let's assume that for n parts of the system we specify a mapping from states of those parts to numerals. If we specify an ordering of these computationally relevant parts of the system, then we can specify the relevant total states of the system with an n-tuple of these interpreted states, $<s_1, s_2,\ldots, s_n>$. We will use 's_i', 's_j' and 's_k' to refer to the i-th, j-th and k-th elements of such an n-tuple of computationally relevant states of P̲ (where i, j, k $\leq$n). Since these n-tuples will indicate our values, we suppose that for any two such n-tuples and for any i, if $s_i \neq s_i'$, then it is not physically possible for P̲ to be in both s_i and s_i' at the same time; thus the state of P̲ at any time can be specified with at most one n-tuple. So let's assume that we have a set S of all and only the states of P̲ that are elements of these n-tuples and a 1-1 mapping from S into a set N of numerals,

$$f_R : S \rightarrow N.$$

This mapping allows us to specify the interpreted state of P̲ at a time with an n-tuple of numerals, $<n_1, n_2,\ldots, n_n>$, i.e., $<f_R(s_1), f_R(s_2),\ldots, f_R(s_n)>$. Our system P̲ can be considered a machine, then, if we have f_R and a 1-1 mapping from N into a set of integers,

$f_M : N \rightarrow I,$

such that the following condition holds in virtue of physical laws that apply to the system:

(R) For some i, j, k $\leq$ n, all s_i and s_j are such that if *P* is in s_i and s_j, then in some (physically possible) circumstances, *C*, *P* will go into a state s_k' such that

$f_M(f_R(s_i)) + f_M(f_R(s_j)) = f_M(f_R(s_k'))$.

Of course, *P* would not be a useful adding machine unless some other conditions held as well:

(1) There are a large number of n-tuples of physical states of *P* for which f_R is defined, so that a large number of sums may be calculated.

(2) It is easy to tell which computationally relevant states *P* is in at any particular time, and it is easy to tell when circumstances *C* come about.

(3) The functions f_M and f_R are easy to compute. We want to be able to read off the mathematical results from the physical states easily.

(4) For many pairs of integers, <i, i'>, it is easy to bring it about that *P* is in the states $s_i = f_R^{-1}(f_M^{-1}(i))$ and $s_j = f_R^{-1}(f_M^{-1}(i'))$ and then to bring about circumstances *C*. That is, we would like to be able to

"initialize" and "start" the system easily.

(5) The system $\underline{P}$ is portable, convenient and inexpensive.

Any of the cheaper hand calculators are examples of such systems. For these, S would include the states of the memory registers. The initialization and start are accomplished just by pressing the right buttons. The computationally relevant states are indicated by the digital display which also indicates the appropriate mapping f_R into numerals, and the mapping f_M is just the standard interpretation of the numerals. Everything is so convenient that the user of the calculator need only think about numbers and the buttons on the calculator; everything else takes care of itself.

It will be convenient to describe such computing systems as <u>physical</u> realizations of <u>abstract</u> machines. In this case, we can think of the abstract machine as a function

$$\underline{M}_{\mathcal{A}} : M \rightarrow M,$$

where M is the set of n-tuples of nuMerals that we can use to specify the interpreted state of $\underline{P}$. The function $\underline{M}_{\mathcal{A}}$ maps each n-tuple of M with elements n_i, n_j, into an n-tuple of M with element n_k such that $f_M(n_i) + f_M(n_j) = f_M(n_k)$. We can define a partial encoding function for $\underline{M}_{\mathcal{A}}$,

$$e: I \times I \rightarrow M.$$

This function maps pairs of integers, <i, i'>, into n-tuples of M such that $n_i = f_M^{-1}(i)$ and $n_j = f_M^{-1}(i')$. The values of the other elements of each n-tuple, if any, can be arbitrarily specified in order to ensure that each pair of integers is mapped into a unique n-tuple in M. The decoding function for $\underline{M}_{\mathcal{A}}$,

$$d\colon\ M \rightarrow I,$$

maps every n-tuple of M into the integer $f_M^{-1}(n_k)$. Thus the composition

$$d \circ \underline{M}_{\mathcal{A}} \circ e$$

is a partial function that maps pairs of integers into their sums. The abstract machine $\underline{M}_{\mathcal{A}}$ is physically realized by $\underline{P}$ because we have the mapping f_R which associates the physical states of $\underline{P}$ with numerals, and the condition (R) which guarantees an appropriate correspondance between changes in physical state and the mapping $\underline{M}_{\mathcal{A}}$.

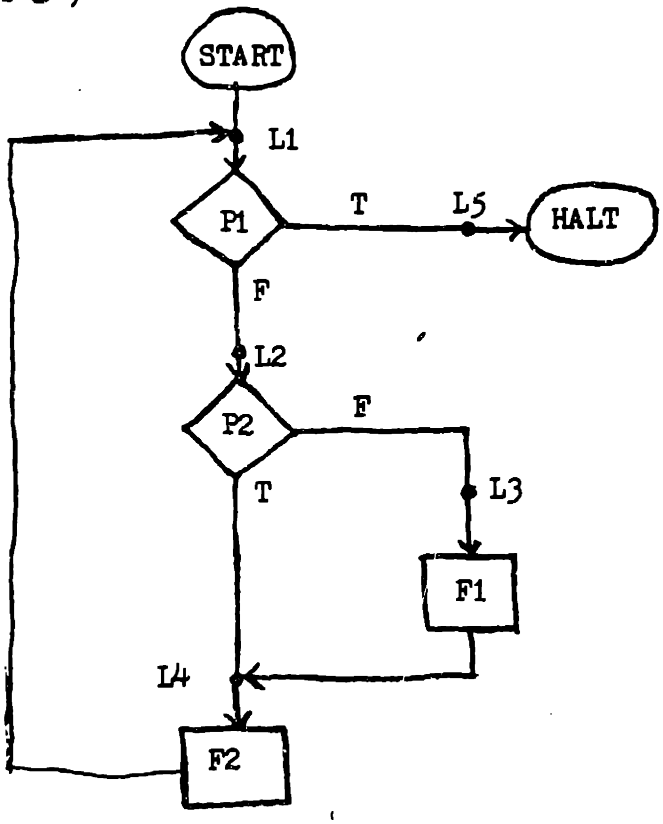
It should be noted that this is an example of something that would, on the present account, count as an adding machine; it is not a specification of what is *required* for something to be an adding machine. The example is intended only to give a simple illustration of how the theory of computation that we will present handles a familiar case.

3. A simple programming language

3.1. Introduction

We will now provide a theory of computation that covers not only simple adding machines but also macnines that can execute complex programs. We will define a language, "SPL," that can represent any program that can be represented by a simple flowchart. Our flowcharts will each have one start circle, at least one halt circle, operation boxes, and test boxes. Thus, if 'F1' and 'F2' are operation names and 'P1' and 'P2' are test names, (FLOWCHART $\mathcal{E}$) represents a simple program:

(FLOWCHART $\mathcal{E}$)



In the programming language SPL, which was designed by Scott(1967), this program is written as follows:

(PROGRAM $\mathcal{E}$)

START: GOTO L1

```
L1:  IF P1 GOTO L5 ELSE GOTO L2
L2:  IF P2 GOTO L4 ELSE GOTO L3
L3:  DO F1 GOTO L4
L4:  DO F2 GOTO L1
L5:  HALT
```

When interpreted in the obvious way, this list of instructions represents the same program as is diagrammed above. Of course, we do not know how to execute the program until we know the intended interpretation of the operation and test names. We will first provide a syntactic definition of SPL.

3.2. The syntax of SPL

I. Vocabulary

a. The following nine symbols we will call the *basic symbols*:

```
START    :      ;
GOTO     ELSE   DO
HALT     THEN   IF
```

B. The *labels*:

L1, L2, L3,...

C. The *operation* (or *function*) *symbols*:

F1, F2, F3,...

D. The *test* (or *predicate*) *symbols*:

P1, P2, P3,...

II. Formation rules

A. The *instructions*:

(i) If L is a label, then ⌜START: GOTO L⌝ is a *start* *instruction*.

(ii) If L and L' are labels and F is a function symbol, then ⌜L: DO F; GOTO L'⌝ is an *operation instruction*.

(iii) If L, L' and L'' are labels and P is a predicate symbol, then ⌜L: IF P THEN GOTO L' ELSE GOTO L''⌝ is a *test instruction*.

(iv) If L is a label, then ⌜L: HALT⌝ is a *halt instruction*.

B. A *program* is a finite set of instructions containing exactly one start instruction and containing for any label in the set exactly one instruction that begins with that label.

III. Additional nomenclature

We will uses '$\underline{L}$' to refer to the set of labels, '$\underline{F}$' to refer to the set of operation or function symbols, '$\underline{P}$' to refer to the set of test or predicate symbols, and '$\underline{A}$' to refer to the set of programs.

3.3. Interpretations of SPL

We will say that an *interpretation of SPL* is a triple <e,$\underline{M}$, d>, where $\underline{M}$ is a machine, e is an encoding function whose range is the domain of $\underline{M}$, and d is a decoding function whose domain is the range of $\underline{M}$. A *machine* is a partial function $\underline{M}$ defined over $\underline{F} \cup \underline{P} \cup \underline{A}$. For elements F_i of the set $\underline{F}$ of function symbols, $\underline{M}(F_i)$ is a computable partial function,

$$\underline{M}_{F_i} : M \rightarrow M,$$

where M is a set of n-tuples of symbols (the memory set). For elements P_i of the set $\underline{P}$ of predicate symbols, $\underline{M}$(Pi) is a computable partial function,

$$\underline{M}_p : M \rightarrow \{ \text{'T'}, \text{'F'} \}.$$

Such an interpretation of the function and predicate symbols of SPL will indicate what operations should be performed to execute any program in SPL, and hence also the values of $\underline{M}$ for elements of $\underline{A}$, i.e., for programs. The former point will be illustrated when we consider a particular interpretation of SPL; we turn now to the latter point. We want to specify the values of $\underline{M}$ for programs.

The values of $\underline{M}$ for arguments that are elements of $\underline{A}$ will be specified according to the values of $\underline{M}$ for the function and predicate symbols. A <u>completed</u> <u>computation</u> by a program π on a machine $\underline{M}$ is a finite sequence,

$$L_1, m_1, L_2, m_2, \ldots, L_n, m_n,$$

such that:

(1) $L_1, L_2, \ldots, L_n \in \underline{L}$;

(2) $m_1, m_2, \ldots, m_n \in M$;

(3) L_1 is the label that occurs in the start instruction of π;

(4) L_n is the label that occurs in some halt instruction of π;

(5) For every label L_i that occurs in the computation,

where 1<i, L_i is some label L such that exactly one of the following conditions hold:

(a) For some $F \in \underline{F}$ and $L' \in \underline{L}$, the instruction $\ulcorner$L: DO F GOTO L'$\urcorner$ is in π, in which case $L_{i+1}=L'$ and $m_{i+1}=\underline{M}_F(m_i)$;

(b) For some $P \in \underline{P}$ and L', $L'' \in \underline{L}$, the instruction $\ulcorner$L: IF P GOTO L' ELSE GOTO L''$\urcorner$ is in π, in which case $m_{i+1}=m_i$ and either $\underline{M}_P(m_i)=$'T' and $L_{i+1}=L'$ or else $\underline{M}_P(m_i)=$'F' and $L_{i+1}=L''$.

For each program π in SPL, then, $\underline{M}(\pi)$ is the partial function,

$$\underline{M}_\pi : M \rightarrow M,$$

where $\underline{M}_\pi(m)=m'$ if there is a completed computation by π on $\underline{M}$, $<L_1, m_1, L_2, m_2, \ldots, L_n, m_n>$, such that $m_1=m$ and $m_n=m'$, and is undefined otherwise. The value of $\underline{M}_\pi(m)$ will be undefined if an operation instruction,

L_i: DO F GOTO L',

is reached where $\underline{M}_F(m_i)$ is undefined, or if a test instruction,

L_i: IF P THEN GOTO L' ELSE GOTO L'',

is reached where $\underline{M}_P(m_i)$ is undefined, or if the program gets caught in an infinite loop and never reaches a halt instruction despite the fact that its function and predicate symbols are defined. So for elements π of the set $\underline{A}$ of programs of SPL,

$\underline{M}(\pi)$ is a partial function,

$$\underline{M}_{\pi} : M \rightarrow M.$$

A *computation* by π on $\underline{M}$ is either a completed computation by π on $\underline{M}$ or else an infinite sequence,

$$L_1, m_1, L_2, m_2, \ldots,$$

such that clauses (1), (2), (3) and (5) of the definition of a completed computation hold. A computation by π on $\underline{M}$ that is not undefined at any step but is not a completed computation will be called a *non-terminating computation*. We will use the following notation,

$$L_1, m_1, L_2, m_2, \ldots (L_n, m_n),$$

to represent a computation that is either non-terminating or else completed and ending with m_n.

If the value of a machine $\underline{M}$ is defined for every function and predicate symbol in a program π, we will say that π is interpreted by $\underline{M}$. If a program π is interpreted by a machine $\underline{M}$, we will call the program π under the interpretation $\underline{M}$ the "$\underline{M}$-program π" or "$\underline{M}$-π".

Although the completed computations of any program (if there are any) are determined by the specification of a machine $\underline{M}$, we do not know what the computations are computations of until we specify the encoding and decoding fuctions. The encoding function maps some set X (the input set) into M (the memory set of $\underline{M}$),

$$e: X \rightarrow M.$$

And the decoding function maps M into some set Y (the output set),

$$d: M \rightarrow Y.$$

The <u>function</u> <u>computed</u> <u>by</u> <u>program</u> π (under the interpretation <e, $\underline{M}$, d>) is the partial function from X into Y,

$$d \circ \underline{M}_{\pi} \circ e.$$

Now we can specify a particular interpretation of SPL and consider what function the program $\mathcal{E}$, presented above, computes under that interpretation.

3.4. An interpretation of SPL

Let our input set X be the set of pairs of positive integers;

$$X = I \times I, \text{ where } I = \{1, 2, 3, \ldots\}.$$

Our memory set M will be the set of pairs of arabic numerals standardly used to represent the positive integers;

$$M = N \times N, \text{ where } N = \{'1', '2', '3', \ldots\}.$$

Let's call the 1-1 function that maps integers into these standard numerals, "f_M,"

$$f_M: I \rightarrow N.$$

The encoding function is the mapping that associates each <u>pair</u> of integers with their standard representations,

$$e: X \rightarrow M, e(\langle x, y\rangle) = \langle f_M(x), f_M(y)\rangle.$$

Now we can define a machine $\underline{M}$:

I. The operation or function symbols

$\underline{M}_{F1}$: $M \rightarrow M$, $\underline{M}_{F1}(\langle x, y\rangle) = \langle y, x\rangle$.

$\underline{M}_{F2}$: $M \rightarrow M$, $\underline{M}_{F2}(\langle x, y\rangle) = \langle f_M(f_M^{-1}(x) - f_M^{-1}(y)), y\rangle$.

II. The predicate or test symbols

$\underline{M}_{P1}$: $M \rightarrow \{ 'T', 'F' \}$,

where $\underline{M}_{P1}(\langle x, y\rangle)='T'$ if $x=y$ and otherwise $\underline{M}_{P1}(\langle x, y\rangle)='F'$;

$\underline{M}_{P2}$: $M \rightarrow \{ 'T', 'F' \}$,

where $\underline{M}_{P2}(\langle x, y\rangle) = 'T'$ if $f_M^{-1}(x) > f_M^{-1}(y)$ and otherwise

$\underline{M}_{P2}(\langle x, y\rangle) = 'F'$.

III. The programs

For every program π,

$\underline{M}_{\pi}$: $M \rightarrow M$,

where $\underline{M}_{\pi}(x) = y$ if there is a completed computation by π on $\underline{M}$, $\langle L_1, m_1, L_2, m_2, \ldots, L_n, m_n\rangle$, such that $x=m_1$ and $y=m_n$;

and otherwise $\underline{M}_{\pi}(x)$ is undefined.

The output set Y is the set of integers. The decoding function maps M into Y,

$$d: M \rightarrow Y, \; d(\langle x, y\rangle) = f_M^{-1}(x).$$

Now we have specified an interpretation, <e, $\underline{M}$, d>, for SPL.

3.5. An example

Under the interpretation just specified, Program $\mathcal{E}$, which was presented above, expresses the familiar Euclidean algorithm for computing the greatest common divisor of two positive integers. The program was the following,

```
(PROGRAM ℰ )
    START:  GOTO L1
    L1:  IF P1 GOTO L5 ELSE GOTO L2
    L2:  IF P2 GOTO L4 ELSE GOTO L3
    L3:  DO F1 GOTO L4
    L4:  DO F2 GOTO L1
    L5:  HALT.
```

(Also see the Flowchart $\mathcal{E}$, above.) It is clear how to go about executing such a program now that the interpretation has been specified. We can render the program into a corresponding set of instructions in English quite easily. The memory set contains pairs of numerals, but given our encoding function we can take them to represent pairs of integers; it is less cumbersome to describe the algorithm in terms of operations on numbers than it is to describe it in terms of operations on symbols. So consider any pair of positive integers. Under the interpretation <e, $\underline{M}$, d>, we proceed according to Program $\mathcal{E}$ as follows:

(1) Check to see if the two integers are the same. If they are, proceed to step (5); otherwise, proceed to step (2).

(2) Check to see if the first integer is larger than the second. If it is, proceed to step (4); otherwise, proceed to step (4).

(3) Reverse the two integers. Proceed to step (4).

(4) Subtract the second integer from the first, and now consider the result and the second integer of the original pair (in that order). Proceed to step (1).

(5) Stop. The decoding function tells us that the second integer is the result.

Now that it is clear how *we* would carry out the program, we will explain what it is for a *program* to compute this program, and then what it is for a *physical system* to do so.

4. Program computation

For the moment we will consider only programs of SPL. We will consider how our account can be extended to other languages in §9, below. We can think of an $\underline{M}$-program as a specification of a certain partial function, namely, the partial function which is the value of $\underline{M}$ for the program taken as argument. An $\underline{M}$-program also specifies a set of computations. We will say that a program π on a machine $\underline{M}$ with memory set M is *computed* or *executed* by a program π' on a machine $\underline{M}'$ with memory set M' just in case there is a 1-1 mapping,

$$g: M \rightarrow M',$$

such that for every computation, c, by π on $\underline{M}$,

$$L_1, m_1, L_2, M_2, \ldots (L_n, m_n),$$

there is a computation, c', by π' on $\underline{M}'$,

$$L'_1, m'_1, L'_2, m'_2, \ldots (L'_q, m'_q),$$

such that:

(i) $g(m_1) = m'_1$;

(ii) if c is completed so is c', in which case $g(m_n)=m'_q$, and

(iii) each member of the sequence,

$$g(m_1), g(m_2), \ldots (g(m_n)),$$

occurs in order (but not necessarily consecutively) in c'.

Thus, if an $\underline{M}$-program π is computed by an $\underline{M}'$-program π', then there is a 1-1 mapping g such that

$$g^{-1} \circ \underline{M}'_{\pi'} \circ g = \underline{M}_{\pi}.$$

We will sometimes speak of part of a program computing or executing some other program, in which case the above definition is to be applied to that part of a program as if it were a program in itself. Notice that if we allowed that for some m_1, $m_2 \in M$ such that $m_1 \neq m_2$, $g(m_1)=g(m_2)$, then even if $\underline{M}_{\pi}(m_1) \neq \underline{M}_{\pi}(m_2)$, there will be no function h such that $h(\underline{M}'(g(m_1)))=h(\underline{M}'(g(m_2)))$. Requiring that g be 1-1 guarantees that this situation will not arise and simplifies our theory without any substantial loss of descriptive power.

It is clear that computation, defined in this way, is reflexive and transitive; that is,

> (a) For every machine *M*, every *M*-program computes itself, and
>
> (b) For any machines, *M*, *M*', *M*'', and programs π, π', π'', if the *M*'-program π' computes the *M*-program π, and the *M*''-program π'' computes the *M*'-program π', then the *M*''-program π'' computes the *M*-program π.

These properties hold for computation by programs, but not, as we will see, for computation by physical systems.

We can now give a similar account of what it is for a physical system to compute an interpreted program. A program on a machine *M* with memory set M is (*physically*) *computed* by a physical system *P* (under the realization function f_R) just in case there is a 1-1 mapping,

$$f_R : M \rightarrow S_M ,$$

where S_M (the memory set of *P*) is a set of physical states of *P* such that,

> (S) For any states s and s' in S_M, if it is physically possible for *P* to be in both s and s' at the same time, then $s \neq s'$.

and the following condition holds in virtue of physical laws applying to the system:

> (R) In certain (physically possible) circumstances, *C*,

for every computation by π on $\underline{M}$,

$L_1, m_1, L_2, m_2, \ldots (L_n, m_n)$,

if $\underline{P}$ is in a state $f_R(m_1)$ then it will go into $f_R(m_2)$,

and then into $f_R(m_3)$, and so on, until it goes into

$f_R(m_n)$ if the computation is completed.

Notice that it follows from these definitions that a physical system (physically) computes an $\underline{M}$-program π if it (physically) computes any $\underline{M}'$-program π' that computes the $\underline{M}$-program π.

If a program π computes some function f under an interpretation <e, $\underline{M}$, d>, and $\underline{P}$ (physically) computes the $\underline{M}$-program π under f_R, then we will say that $\underline{P}$ (physically) computes $\underline{f}$. Thus, if $\underline{P}$ computes some $\underline{M}$-program π, it computes the function $\underline{M}_\pi$, since π computes $\underline{M}_\pi$ under the interpretation <e, $\underline{M}$, d>, where e and d are the identity function on M. So a physical system computes $\underline{M}_\pi$ (as a program does also) just in case it computes the following program:

```
(PROGRAM Γ )
    START:  GOTO L1
    L1:  DO F1 GOTO L2
    L2:  HALT,
```

where $\underline{M}'_{F1} = \underline{M}_\pi$. (We will call programs of this form one-step programs.) A completed computation of this program on $\underline{M}'$ would be a sequence,

$$L_1, m_1, L_2, m_2,$$

where L_1=L1, L_2=L2, and $m_2 = M'_{F1}(m_1) = M'_{\Gamma}(m_1) = M_{\Pi}(m_1)$. So a physical system that (for every completed computation by Γ on M') goes into state $f_R(m_2)$ because it was in $f_R(m_1)$ whenever some specifiable circumstances, C, come about, computes this M'-program. Notice that if a complete computation by π on M is a longer sequence, then a physical system could compute the M'-program Γ (and hence M_{π}) but not compute the M-program π. It is easy to see, for example, that there could be physical systems that compute $M_{\mathcal{E}}$ under some f_R but do not compute the M-program $\mathcal{E}$ (discussed above) under f_R, but use some other program instead. We also have the result that the simple calculator with which we began computes the function $M_{\mathcal{R}}$ (which was specified in §2, above); it may compute this function by computing some complex program, but this need not be the case.

5. Compilers and interpreters

A compiler (or translator) is an M-program Θ that, under some interpretation <e, M, d>, computes a function,

$$\mathcal{C}: A \rightarrow A,$$

where for any program π', $\mathcal{C}(\pi')$ is a program π'' such that, for some M', M'', the M''-program π'' computes the M'-program π'. We will say that such an M-program "compiles M'-programs into M''-programs," and that it "compiles M' into M''." Following Clark and Cowell (1976, pp.24-25) we will say that M'' simulates

$\underline{M}'$ if there is a compiler that compiles $\underline{M}'$ into $\underline{M}''$, and that two machines are equivalent if each simulates the other.

If we have a device which executes $\underline{M}$-programs, then, and we want to compute some $\underline{M}'$-program π' on it, we could use an $\underline{M}$-program Θ which compiles $\underline{M}'$ into $\underline{M}$ (under some interpretation $\langle e, \underline{M}, d\rangle$) to compile an $\underline{M}$-program which computes the $\underline{M}'$-program π'. We could then compute the $\underline{M}$-program π on our device. In fact, we could write a "compile-and-go" program which, in effect, compiles an $\underline{M}'$-program into an $\underline{M}$-program and then executes the $\underline{M}$-program. Consider, for example, the following $\underline{M}$-program Υ,

```
(PROGRAM Υ )
        START:  GOTO L1
        L1:  DO F1 GOTO L2
        L2:  DO F2 GOTO L3
        L3:  HALT,
```

where $\underline{M}_{F1}=\underline{M}_{\Theta}$ and $\underline{M}_{F2}: M \to M$, $\underline{M}_{F2}(m)=\underline{M}_{d(m)}(m)$, where Θ is a compiler of $\underline{M}'$ into $\underline{M}$ under $\langle e, \underline{M}, d\rangle$. We assume that the execution of the compiling step, L1, does not transform the memory in any respect that is relevant to the computation of the second step, L2; that is, for all completed computations of on $\underline{M}$,

$$L_1, m_1, L_2, m_2, L_3, m_3,$$

$\underline{M}_{F2}(m_1)=\underline{M}_{F2}(m_2)=m_3$. Intuitively, the encoding of the $\underline{M}'$-program and the encoding of the compiled $\underline{M}$-program do not intrude on the space in memory that holds the values relevant to the execution of the second step of the program.

Notice that if the memory set of $\underline{M}$ is finite, and there are an infinite number of (interpreted) $\underline{M}$-programs no two of which compute each other, then no $\underline{M}$-program will be able to compile all $\underline{M}'$-programs in the way we have indicated. If we allowed our input to be read into the memory set in a "stream," as a series of memory states, then it might be possible to have an $\underline{M}$-program that compiled infinitely many different $\underline{M}'$-programs.[2] But for the present we will stick to encodings and decodings of single memory states.

If the memory set of $\underline{M}$ is finite, then, and so d[M] is finite (where d is the decoding function of our compiler, Θ), then we can write an $\underline{M}$-program which computes $\underline{M}$-$\mathcal{I}$ by actually executing all the instructions of the appropriate compiled program represented in d[M]. We impose an ordering on the set d[M] so we can list the compiled programs as follows,

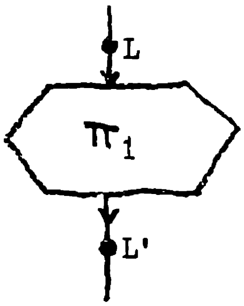
$$\pi_1, \pi_2, \ldots, \pi_n.$$

We will use the flowchart enclosure of (FLOWCHART π_1) as an abbreviated notation for the flowchart of program π_1, except that the start enclosure of π_1's flowchart is removed and L is the label that occurs in the start instruction of π_1, and all

---

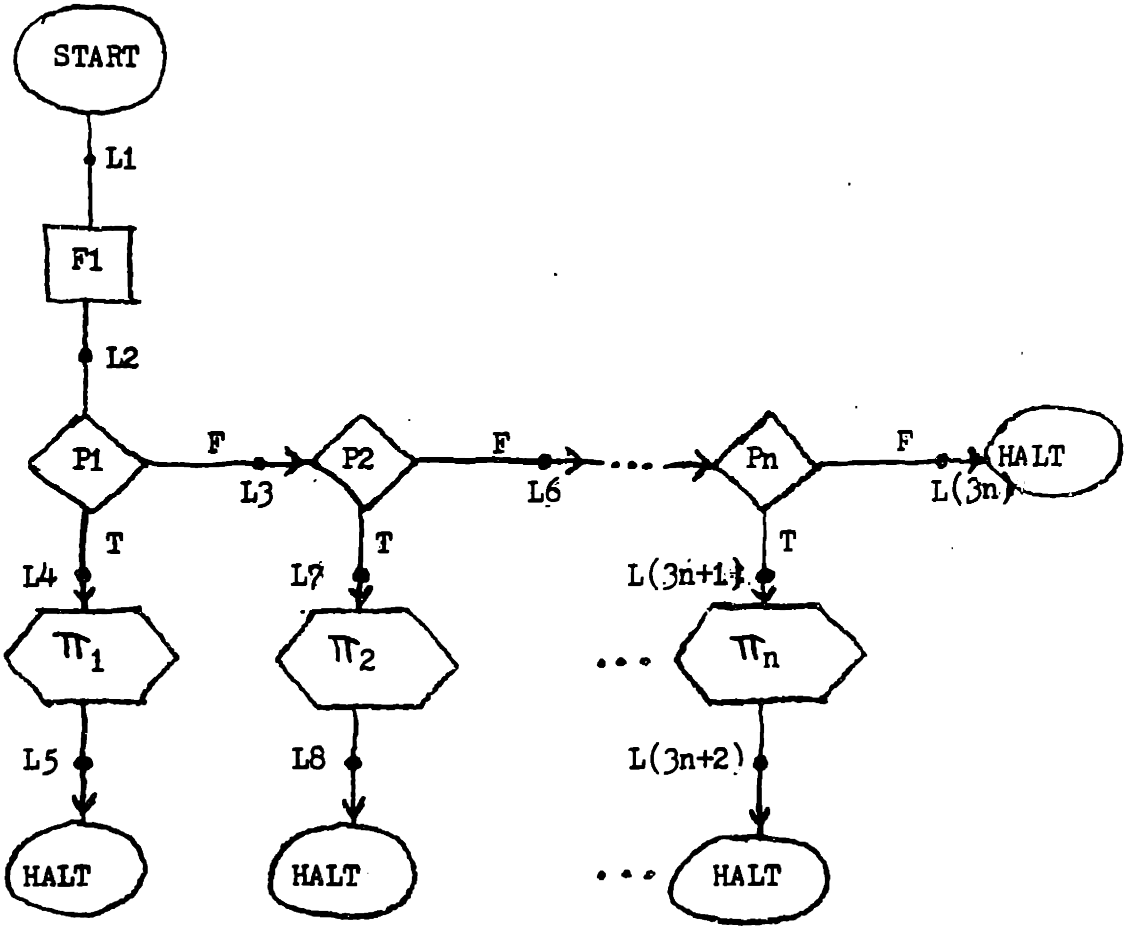
[2] See Clark and Cowell, 1976, ch.5, for an extension of this account to machines with input and output streams.

(FLOWCHART π_1)



halt enclosures are removed and the lines leading to them are connected to L'. Then a more sophisticated "compile-and-go" program can be represented as in (FLOWCHART K).

(FLOWCHART K)



We assume that none of the labels L1, L2, L3,..., L(3n+2) occur inside any of the program enclosures, and that all operation and test symbols occurring in these enclosures are interpreted by $\underline{M}$. The interpretation of the function symbol F1 and the test symbols P1, P2,..., Pn is as follows:

$$\underline{M}_{F1} = \underline{M}_{\Theta},$$

and for every predicate symbol Pi, $1 \leq i \leq n$, the value of $\underline{M}$(Pi) is the function

$$\underline{M}_{Pi} : M \rightarrow \{ \text{'T'}, \text{'F'} \},$$

where $\underline{M}_{Pi}(m) = \text{'T'}$ if $m \in \{ x \mid \langle x, \pi_i \rangle \in d \}$ and otherwise $\underline{M}_{Pi}(m) = \text{'F'}$. It should be clear that $\underline{M}$-κ computes $\underline{M}$-Υ by executing the steps of the appropriate compiled programs.

This last program raises the interesting point that the compiling step really is not needed if we have tests sufficient to distinguish the encodings of the different $\underline{M}$'-programs. If we have such tests, then we can just use the encodings of the $\underline{M}$'-programs to cue the execution of the appropriate $\underline{M}$-program π_i. A (<u>software</u>) <u>interpreter</u> or <u>software simulation</u> of a machine $\underline{M}$' is an $\underline{M}$-program Λ that works in this way; it computes $\underline{M}$'-programs without compiling them as $\underline{M}$-κ does. Instead, an encoding of the $\underline{M}$'-program is executed step-by-step in the course of executing $\underline{M}$-Λ. If none of the $\underline{M}$-program is compiled we say that $\underline{M}$-Λ is a <u>pure simulation</u> of $\underline{M}$' on $\underline{M}$. Unfortunately, it is beyond the scope of this discussion to describe any compilers or interpreters in detail, but they are of interest and so it is important to have a rough idea of what they are. We will consider them again below.

6. Direct systems

We will distinguish two sorts of physical systems that compute programs in the sense specified above: "direct systems" and "program using systems." A "direct" or "hardwired" system is a physical system that computes some <u>M</u>-program π but not by using a representation of the program to control its operation; rather, it operates in the way it does just because of the way it is built. We can be more precise about this once we have characterized program using systems, but the intuitive idea of "direct" computation can be made clear by considering some examples. These wil be examples of systems that compute programs but clearly do not <u>use</u> programs to govern their operation.

6.1. First example

Let's consider first a naturally occurring system that computes some interesting program without any human interference, since it is such natural systems that we are most interested in being able to describe. So consider some naturally occurring physical system whose operation corresponds to the behavior of some interesting function. Since physicists like to find functions which correspond to nice descriptions of physical systems, they are well equipped to provide us with the sort of example we are looking for. Consider Newton's law of gravitation, for example,

$$F = G m_1 m_2 r^{-2} .$$

This equation holds true (at least to a good approximation) of any two masses, m_1 and m_2; they will attract each other with a force F which is inversely proportional to the square of the distance r between them. If we consider as our two masses the sun and a planet, then we can use this formula and ignore the relatively small effects of other masses upon the system. And if we take a planet which is not too hard to see but has a rather eccentric orbit, then we can use the system to compute the values of F for a range of r, since G, m_1 and m_2 are constant. So consider the system consisting of the sun and Mars, whose orbit was eccentric enough to lead Kepler to his first law. Let's describe a program that this system actually computes just to make quite clear how it could be done.

To simplify the example we will only work on finding a program that the system computes when Mars is getting further away from the sun, travelling from its perihelion to its aphelion, and we will keep our figures in a range that allows us to use standard units of measurement. At any point in its orbit Mars (with mass ♂) will be at a distance r from the sun (with mass ⊙), and acted upon by a force F according to Newton's law. So let our memory set M be the set of pairs of standard decimal numerals, <r, F>, and let our encoding and decoding functions be the standard interpretation of these pairs of numerals as pairs of numbers. Let f_M be the function which maps each numeral into the rational number it standardly represents. Now we can describe the system as computing the following M-program,

```
(PROGRAM ℕ )
    START:  GOTO L1
    L1:  DO F1 GOTO L2
    L2:  HALT,
```

where $\underline{M}$ is defined as follows:

$$\underline{M}_{F1} : M \rightarrow M,$$

where,

$$\underline{M}_{F1}(\langle r,F\rangle)=\langle f_M^{-1}(f_M(r)+1),\ f_M^{-1}(Gm_{\odot}m_{♂}(f_M(r)+1)^{-2})\rangle$$

if $2.1 \times 10^{11} < f_M(r) < 2.4 \times 10^{11}$, and is undefined otherwise. This program successively increments r by one and then computes the new value of F. Now we can specify the realization function f_R as follows,

$$f_R : M \rightarrow S,\ f_R(\langle r, F\rangle)=\langle s, s'\rangle,$$

where s is the state of Mars' being $f_M(r)$ meters away from the sun, and s' is the state of Mars' being acted upon by a gravitational force of $f_M(F)$ Newtons. If Mars is on its trip from perihelion to aphelion, and if no other significant forces are applied to the system, then for every computation by ℕ on $\underline{M}$,

$$L_1, m_1, L_2, m_2,$$

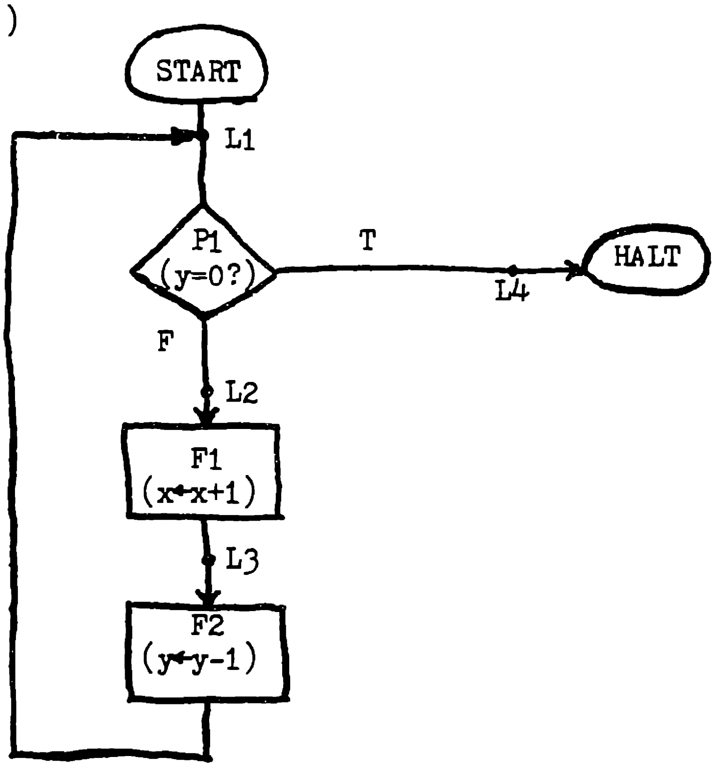
if the system is in some state $f_R(m_1)$ (such that $\underline{M}_{F1}(m_1)$ is defined) then physical laws that apply to the system determine that it will go into the state $f_R(m_2)$, i.e., $f_R(\underline{M}_{F1}(m_1))$. An astronomer who could determine (quite precisely!) the values of r and F by observation, and who knew, for example, how long it takes for the system to go from some state $f_R(m_1)$ into the next computationally relevant state, $f_R(m_2)$, could use this system to calculate solutions to Newton's law. This would be a wildly impractical scheme, of course, but it could, in principle, be done.

It might seem that although we can describe physical systems as computing some programs in this way, there could not be a physical system that directly computed extremely complicated programs. But in fact, every program that can be computed by a program using system can be computed directly by a system that clearly does not use any representation of the program to govern its operation. Our next example will suggest a method for building an electronic circuit that can compute any particular SPL program directly.

6.2. Second example

Consider an SPL program with a loop, like the following $\underline{M}$-program $\mathcal{A}$ that computes the sums of pairs <x,y> of positive integers. (The interpretation of the function and predicate symbols is indicated parenthetically in each box.) The challenge in finding a system that can compute a program like this one is

(FLOWCHART A)



to find a system that will make the appropriate number of loops for any given input. One way of getting the system to do this is to control its operation with a representation of the program, but there are other ways. One is to use what Von Neumann (1958) calls "plugged control." That is, we could just take electronic circuits, electronic "boxes," which perform the operations and tests corresponding to F1, F2, and P1 (under some realization function f_R) and "plug" them together in the way the flowchart indicates. Thus, the electronic box corresponding to F1 in our *M*-program might be a circuit that would take two inputs, x and y, whose voltages are paired with integer values by f_R; it would (after some delay) provide two outputs, one with the voltage of one of the inputs and a voltage which (under f_R) represents the other's integer value plus one. We would connect the output of this box to the input of the F2 box, and so on until the whole "plugged circuit" is connected in the way indicated by the flowchart. The box corresponding to P1 would pass its input to one output (the 'T' output) if the voltage of its y-input represented 0 under f_R, and otherwise would pass the input to the other output point (the 'F' output). It is not difficult to design electronic circuits which (under some f_R) behave in the way we would require these to behave. We will call a system that is constructed from electronic boxes in this way a "block diagram system."[3]

It is clear that we can describe these block diagram systems as computing the corresponding interpreted programs. Consider again the block diagram system corresponding to our M-program $\mathcal{A}$. Given some input, $f_R(m_1)$, physical laws applying to the electronic circuits determine that the system will go into the next computationally relevant state, in which an appropriate signal has been generated at one of the outputs of the box P1. If the signal appears at the 'F' output, the next computationally relevant state, $f_R(m_3)$, will be one in which a signal has been generated at the output of box F1, and so on until we get our final output at the 'T' branch of P1, if ever. This final output will be a voltage that represents the sum of the integer values

[3] I have taken this term from Dertouzos et al., 1972, where it is used to describe similar "plugged control" systems.

of our input.

Notice that our various electronic boxes can share circuitry so long as we have the right relations between inputs and outputs. And notice that even if one of our electronic boxes were itself a program using system, the whole plugged system, the system that computes the M-program $\mathcal{A}$, would not be a program using system since its execution of the M-program $\mathcal{A}$ is brought about by the wiring of the circuit, not by the control of some representation of formulae of our programming language in (internal or external) memory.

Although any program that can be computed by a program using system can be computed by a direct system, i.e., by a system that clearly does not make use of the representation of a program, building direct systems to execute non-trivial programs is usually impractical given the current state of technology. For most purposes it is much more practical to use a program using system which, in effect, sets up the right "wiring" in the course of the computation.

7. Program using systems

A "program using system" or "program system" is a physical system whose computation of some M-program π is controlled by a representation of the program π. The distinction between direct and program systems is not just that the program is represented in a program system, since programs may be encoded in the memory set of a direct system like the Mars-sun system. And the distinction is not just that the represented programs have an influence on the computational process in the program using system, since the physical states which are encoded in direct systems also have an influence on the operation of those systems. A program system is distinguished rather by its possession of some "control mechanism" which makes use of the representation of the program in determining what tests and operations will be performed in computing the program. Let's try to be a bit more precise about this.

A program using system that computes $\underline{M}$-π under some f_R uses a representation of the program π to control its operation. Let's define a function r which maps programs into sets of instructions, such that for any π,

$$r(\pi) = \{x \mid x \in \pi \text{ and } x \text{ is neither a start instruction nor a halt instruction}\}.$$

The representation of a program π that a program system uses, then, can be specified with a 1-1 mapping,

$$f_\pi\colon\ r(\pi) \rightarrow S_A,$$

where $r(\pi)$ is non-empty and S_A is a set of physical states of the system $\underline{P}$.

The states in $f_\pi[r(\pi)]$ for any π, i.e., the states in the image of any $r(\pi)$ under the mapping f_π, may be states of some input device such as a tape reader, or they may be memory states of a stored-program system. In either case, the system must be influenced by the image of the appropriate instruction under f at each point in every computation. This influence is brought about through the operation of some "control mechanism" that determines that the right instruction has the right influence at the right time. So when the system executes an instruction I_j, it must make the appropriate transition from $f_R(M_j)$ to $f_R(m_{j+1})$, but also, the control mechanism must ensure that the next state transition will be controlled by $f_{\pi}(I_{j+1})$. In a typical digital computer executing a machine language instruction, it is the state of a certain memory register (viz., the program counter) which determines which instruction will be executed next, and so the contents of this register must be appropriately reset in the execution of each instruction.

So in any program using system, P̲, we can think of there being a set S_C of "control states" such that if P̲ is in some physical state $s_C \in S_C$, then it will execute the instruction paired with some particular state $s_A \in S_A$ by f_{π}, rather than any other instruction. So we can think of the "control mechanism " as determining a partial 1-1 function,

$$C: S_A \rightarrow S_C,$$

so that any particular instruction I is executed only if the system is in a control state $C(f_{\pi}(I))$.

Now we can draw all of these parts of a program using system together in a definition. In a computation by π on $\underline{M}$,

$$L_1, m_1, L_2, m_2, \ldots (L_n, m_n),$$

$L_1, L_2, \ldots (L_n)$, are labels that occur in π; let's use I_1, $I_2, \ldots (I_n)$, respectively, to refer to the instructions in which begin with these labels. We will say that a physical system $\underline{P}$ is a program using system if and only if it "uses" some program π to compute $\underline{M}$-π (for some $\underline{M}$). And we will say that $\underline{P}$ uses π to compute $\underline{M}$-π (under f_R and f_π) if and only if,

(i) there is a 1-1 mapping,

$$f_R : M \rightarrow S_M,$$

where M is the memory set of machine $\underline{M}$ and S_M is a set of physical states of $\underline{P}$ such that:

(S) for any states s and s' in S_M, if it is physically possible for $\underline{P}$ to be in both s and s' at the same time, then $s \neq s'$;

(ii) there is a 1-1 mapping,

$$f_\pi : r(\pi) \rightarrow S_A,$$

where S_A is a set of physical states of $\underline{P}$;

(iii) there is a 1-1 partial function,

$$C: S_A \rightarrow S_C,$$

(the control mechanism) where S_C is a set of physical states of P (the control states), and

(iv) physical laws applying to the system P determine that, in certain (physically possible) circumstances, C, for every computation by π on M,

$L_1, m_1, L_2, m_2, \ldots (L_n, m_n)$,

for any m_i in such a computation, if P is in $f_R(m_i)$ and $f_\pi(I_i)$, then it will go into $f_R(m_{i+1})$ because it is in $f_R(m_i)$ and $f_\pi(I_i)$, and $f_\pi(I_i)$ has this influence because P is in the control state $C(f_\pi(I_i))$.

According to this definition, if P completes a computation of on M, coming finally into $f_R(m_n)$, the last instruction to be executed will be I_{n-1}. So this definition requires that in the operation of the system in circumstances C, the states which are paired by f_π with the instructions $I_1, I_2, \ldots (I_{n-1})$ of any computation by π on M are causally efficacious at the respective points in the computation because of the conĉrol mechanism.

Notice that this set of instructions which are causally efficacious does not include a start instruction or a halt instruction; I_1 is the instruction executed immediately after the start, and I_{n-1} is the instruction before the halt. We used the function r in order to avoid the requirement that these instructions are even represented. Neither start nor halt instructions are ever involved in bringing about the transition from $f_R(m_1)$ to $f_R(m_n)$ in any completed computation; rather they concern what the system is to do at the onset and completion of a computation. "Executing" a start instruction amounts to bringing about the conditions, C, in which P will start executing the program, i.e., start passing through the states $f_R(m_1)$, $f_R(m_2)$, and so on. And we do not need to concern ourselves with what happens to the system when (and if) it ever executes a halt instruction.

Now we can define a direct or hardwired system as a system that computes some program π but for which there is no intelligible, correct description of the system as a program using system that computes π. Thus, we will restrict our application of the term 'direct system' to systems that we could not correctly describe as program using systems. The Mars-Sun system and the plugged-control systems are clear examples of this sort.

We are at last ready to answer the difficult question that we began with: What is it for a system to use a programming language? Or rather, we are now in a position to explain what it is for a system to use SPL, and other programming languages could be handled similarly, as we will point out below.

We will say that a program using system that can compute any (or at least many) *M*-programs "realizes" or "simulates" *M* and "uses *M*-SPL." That is, *P* *realizes* *M* and *simulates* *M* if and only if *P* "uses *M*-SPL." And a physical system *P* *uses* *M-SPL* (under f_R and f) just in case *M* is a machine with memory set m such that:

(i) there is a 1-1 mapping,

$f_R : M \rightarrow S_M$,

where S_M is a set of physical states of *P* such that:

(S) for any states s and s' in S_M, if it is physically possible for *P* to be in both s and s' at the same time, then $s \neq s'$;

(ii) there is a (partial) 1-1 mapping f defined over *A*, where for each $\pi \in$ *A*, the value of $f(\pi)$ (if it is defined) is a function,

$f_\pi : r(\pi) \rightarrow S_A$,

where S_A is a set of physical states of *P*;

(iii) there is a partial 1-1 function,

$C: S_A \rightarrow S_C$,

(the control mechanism), where S_C is a set of physical states of *P* (the control states), and

(iv) for each $\pi \in$ *A* for which $f(\pi)$ is defined, the following condition holds:

(R) physical laws applying to *P* determine that, in certain (physically possible) circumstances, *C*, for every computation by π on *M*,

$$L_1, m_1, L_2, m_2, \ldots (L_n, m_n),$$

for any m_i in such a computation, if *P* is in $f_R(m_i)$ and $f_{\pi}(I_i)$, then it will go into $f_R(m_{i+1})$ because it is in $f_R(m_i)$ and $f_{\pi}(I_i)$, and $f_{\pi}(I_i)$ has this influence because *P* is in control state $C(f_{\pi}(I_i))$.

We would not want to describe a system as a system that uses *M*-SPL if f were everywhere undefined; we assume that it must be defined for a substantial number of programs. It might, for example, be defined for any program that has less than 1000 instructions. Allowing f to be partially undefined makes it possible to describe stored-program systems with finite memories as systems that use *M*-SPL.

A system may simulate a machine *M* by computing a software interpretation of the machine, as was noted above, in §5. In such a case, *P* would typically compute an *M*-program π by executing some other program that is not computed by *M*-π. So we will say that if *P* uses *M*-SPL under f_R and f, and if when it computes any *M*-program π under f_R it is not computing any program that is not computed by *M*-π, then *P* is a *hardware interpretation* or *hardware simulation* of *M*. In such a case. the programming language *M*-SPL is sometimes called the *machine language* of *P*.

Just as it is possible to build a machine to directly compute any interpreted program (when the memory set specified by the interpretation is finite), so it is possible to build a hardware simulation of any machine (with a finite memory set). But this is not practical for most machines. It is generally much more economical to build hardware simulations of machines with a few very simple operations and tests, and then to simulate other "higher level" machines with software interpreters, or just to use a compiler to compile the higher level machines into the machines we can simulate on our hardware.

Such "layers" of machines are commonly used on typical digital computers (though these machines are not typically SPL-machines). This layering is described in the following passages:

> All the time I design programs for nonexisting machines and add: "if we now had a machine comprising the primitives here assumed, then the job is done."...In actual practice, of course, this ideal machine will turn out not to exist, so our next task -- structurally identical to the original one -- is to program the simulation of the "upper" machine...But this bunch of programs is written for a machine that in all probability will not exist, so our next job will be to simulate it in terms of programs for a next lower machine, etc., until finally we have a program that can be executed by our hardware. (Dijkstra, 1972, pp.48-49)
>
> A hardware computer is termed an *actual computer*. A computer that is partially of wholly simulated by software or microprograms is properly termed a *virtual computer*...The virtual computer that a programmer uses when he programs is in fact formed from a *hierarchy of virtual computers*...A language like SNOBOL4 is sometimes implemented by coding the translator and simulation routines in another high-level language such as FORTRAN.

> It would be appropriate to say in such a case that the SNOBOL4 virtual computer is being simulated on the FORTRAN virtual computer, which in turn is being simulated on the operating-system-defined virtual computer, which is itself being simulated on the hardware computer (or on the firmware computer that is being simulated by microprograms on the hardware computer)...The hierarchy does not end with the high-level language implementation. The programs the programmer runs add another level to the hierarchy. (Pratt, 1975, pp.29-32)

Our account makes clear that there must be a definite correspondance between the computations of even the highest-level program and the physical state transitions of the hardware, but as these passages suggest, the correspondance is typically forbiddingly complicated to spell out in detail. Specifying how each interpretation simulates the next highest machine is only one step in describing the correspondance between the highest level and the hardware, and even one such step is usually quite elaborate.

8. Preliminary objections

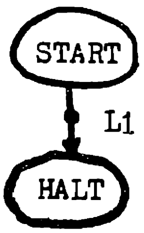
In this section steps will be taken to avoid two likely objections to our account of computing systems, viz., that it is not desciptively accurate in certain cases, and that it makes a distinction between direct systems and program using systems where there really is no distinction. We will consider each of these worries in turn.

8.1. Applications of the framework

As was indicated above, the account of SPL that has been presented here (in §3) is based on Scott(1967) where the relation of this formalism to other more standard formalisms of automata theory is clearly indicated. This is also done in Clark and Cowell(1976). The sort of approach to SPL that we have used here has been extended to other more common and much more complicated programming languages; a formal interpretation of these languages maps well-formed formulae of the languages into functions over sets which include their memory sets, and implementations of the languages are defined accordingly. See Tennant(1976) and Stoy(1977) for introductions to this work and further references. There are other approaches to interpreting programming langauges and defining implementations (cf., e.g., Dijkstra, 1976; Brady, 1977), but there does not seem to be any objection in principle to the approach adopted here.

It is important to recognize that our definitions are not intended primarily to be adequate descriptions of the intended meaning of these terms in ordinary talk, any more than automata-theoretic definitions of machines are intended to capture the ordinary meaning of the term 'machine'. As was pointed out at the beginning of this chapter, it is enormously useful to describe some physical systems as computing systems. However, it is ridiculous to so describe others, as we have seen in our Mars-sun example. In many cases there would be no point to giving computational descriptions of a system. For example, what point is there in describing a physical system as computing the empty program illustrated in (FLOWCHART △), or in describing a program system that only computes a one-step program? We could try to rule out these cases. We could, for example, decide not

(FLOWCHART Δ)



to count one-step program using systems as program using systems. But then, should we allow two-step program using systems? Since there is no theoretically interesting difference between these simple systems and the more complicated ones in which we are interested, no attempt has been made to trim out the "ridiculous" cases.

Along the same lines, it is interesting to note that, on our account, a system that computes the empty $\underline{M}$-program Δ computes the function computed by <u>any</u> $\underline{M}'$-program π' if only there is a 1-1 encoding,

$$g: M' \to M,$$

where M' is the memory set of $\underline{M}'$ and M is the memory set of $\underline{M}$. Consider the function computed by any π' under any interpretation $<e', \underline{M}', d'>$, viz.,

$$d' \circ \underline{M}'_{\pi'} \circ e'.$$

Well, Δ computes this function under some interpretation $<e, \underline{M}, d>$, just in case

$$d \circ \underline{M}_{\Delta} \circ e = d' \circ \underline{M}'_{\pi'} \circ e'.$$

This will be the case if we let

$$e = g \circ e',$$

and,

$$d = d' \circ \underline{M}'_{\pi'} \circ g^{-1} ,$$

since $\underline{M}_{\Delta}$ is the identity function on M.

Computational descriptions are interesting only in certain cases, then. Typically, they are not interesting if the system computes only an $\underline{M}$-program π such that $\underline{M}$-π is trivial. They are likely to be interesting only in those cases where the system computes interpreted programs which involve rather complicated transformations of the memory set of the machine. Since our realization functions are 1-1 mappings,

$$f_R : M \rightarrow S_M ,$$

a system that computes a "complicated" program will always be one that undergoes correspondingly "complicated" state transitions. And we have been assuming that the physical laws applying to the system require it to undergo these changes in the circumstances in which the program is computed, so only certain special systems can correctly be described as computing non-trivial programs.

In any case, these considerations point up the fact that the definitions we have presented here are not really descriptive definitions; they do not purport to capture the"standard" meanings of these terms as they are commonly used. The Mars-sun system would not usually be called a computing system, for example. But the definitions presented above are intended to correspond to standard usage in most of the non-trivial cases. Thus anything that an engineer would want to count as a "hardwired computer" or as an "SPL-using system" or as an "SPL compiler" should count as such under our definitions. The real value of the definitions does not reside in whatever descriptive accuracy they might possess so much as in the fact that they are relatively precise and reconstruct notions that might be of use. Our interest is not in classifying things according to these definitions rather than according to some others, but rather to have fairly clear definitions where there were none before and to use them to illuminate what has been vague or obscure.

8.2. Programs and data

One often hears it said that there is no difference between programs and data, that a program can be considered data and data can be considered a program. Consider the following passages:

> To make the usual distinction between program and data we must divide the information placed initially on the tape into two parts, one part to be called the program and the other part the data. We then think of the program as defining a function, and the data as constituting the argument or arguments of the function, and the computed output as being the function value for those arguments. As so described, the distinction between program and data is purely arbitrary, and this is certainly so from a purely formal point of view. For example, it is arbitrary whether we say a number referred to in a conditional shift of control (branch) command belongs to the program or the data or both, and when the program is the object of computation (e.g., in compiling) the program <u>is</u> the data. Our criterion for distinguishing program from data is an informal, intuitive one: the program is that which, for the most part, directs operations; the data are those items which, for the most part, are operated upon. (Burks, 1963, p.105)
>
> Implicit in the above discussion is a central concept that deserves explicit mention: the equivalence of programs and data. We are accustomed to considering certain kinds of objects in programming as "program" and others as "data." This is often a useful intuitive distinction, but...it is more apparent than real.That

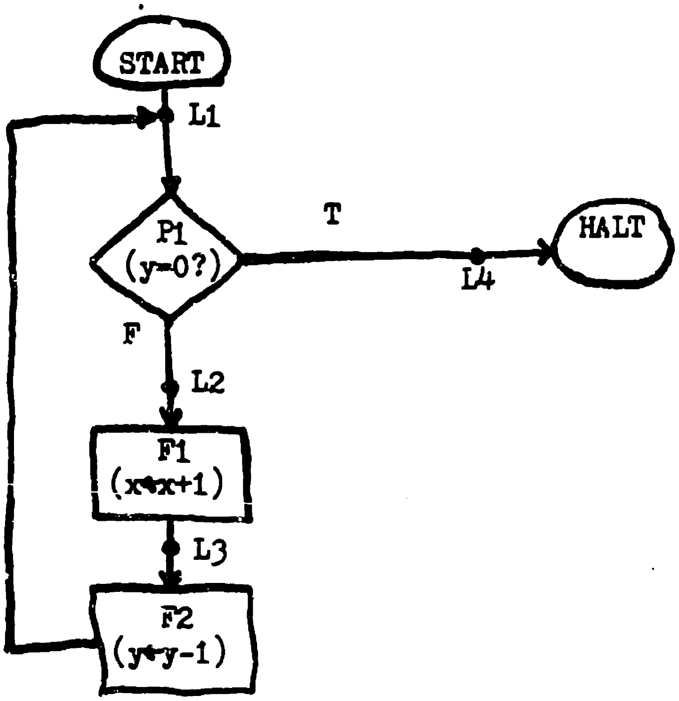
> which is program in one context is likely to become data in another. For example you may write an ALGOL program, but to the ALGOL compiler that program is input data which it is to process. The output data produced by by the compiler is, to you, a program in machine language. You might request execution of this program, but a closer look might convince you that in fact the program is just data to the interpreter used by the executing computer. In the same vein we may always consider the inputs to any program equivalently as data to be processed or as program to be executed. (Pratt, 1975, p.32)

We do not have any stake in any distinction between "data" and "programs" *per se*. Certainly programs that are executed in one computation may be the data in another, as we saw in our discussion of compilers. And an interpreter brings it about that an *M*-program is computed by computing some *M*'-program which, in effect, treats the *M*-program as data. We will even want to extend our account to languages in which we may have self-modifying programs that treat themselves as data. But, aside from the distinction between data and programs as objects, these passages raise an interesting question about computing *systems*. If data can *always* be considered a program, then is every direct system a program system?

It should be clear that, on our account, this is not the case. The criterion for distinguishing a program from data (when they are distinct) is not *how much* they govern the operation of the machine, but the *mechanism* by which they govern operations. To say that a system is a program using system is to say that its operation is controlled in a certain way, a way which we have tried to capture in our definitions.

To illustrate this point, let's see if we can give an equivalent description of the block digram system discussed above (in §6.2.) as a program system. That is, given that this system is a direct system that computes $\underline{M}$-$\mathcal{A}$, as described, can we deduce that it is a program using system that computes $\underline{M}$-$\mathcal{A}$? The direct, block diagram system is a physical system $\underline{P}$ which, under some f_R, computes the sums of pairs of positive integers <x, y> by computing the $\underline{M}$-program $\mathcal{A}$ represented in the flowchart repeated below.

(FLOWCHART $\mathcal{A}$)



We will assume that the memory set M is some large but finite set of pairs of numerals standardly used to represent the positive integers, and so, for example, $\underline{M}_{\mathcal{A}}$ (<'2', '3'>)=<'5', '0'>. Can we deduce from this account an account of the system $\underline{P}$ as a program using system that computes any program?

We come the closest to being able to do so when we try to describe $\underline{P}$ as the limiting case of a program system, i.e., as a one-step program system. So let's try to describe $\underline{P}$ as a program using system which, in effect, treats the data <'2', '3'> as a representation of the following $\underline{M}'$-program:

```
(PROGRAM ø )
    START:  GOTO L1
    L1:  DO F1 GOTO L2
    L2:  HALT,
```

where, letting f_M be the standard interpretation of the numerals,

$$\underline{M}'_{F1} = \underline{M}'_{\phi} ,$$

and,

$$\underline{M}'_{\phi} : M' \rightarrow M', \underline{M}' \ (\langle x,y\rangle) = \langle f_M^{-1}(f_M(x)+f_M(y)), \text{'0'}\rangle.$$

According to our account of program using systems (in §7), the representation of this program ϕ is specified by a 1-1 mapping,

$$f_{\phi} : r(\phi) \rightarrow S_A .$$

In this case, $r(\phi)$= 'L1; DO F1 GOTO L2', so we need exactly one state in $f_{\phi}[r(\phi)]$. The idea, then, is to let $f_{\phi}(r(\phi)) = f_R(\langle\text{'2', '3'}\rangle)$. All we need now is a control state s_C that is responsible for the fact that $f_{\phi}(r(\phi))$ has the influence it does, and our description of $\underline{P}$ as a program using system that computes $\underline{M}'$-ϕ will be complete. But here we run into difficulty; indeed, this is exactly where one would expect to run into difficulty, since it is the way control is managed that distinguishes program using systems from direct systems.

The problem here is that nothing in our description of *P* as a direct system provides us with, or even assures us that there is, an appropriate control state s_c. So it does not follow from our our claim to have a direct system that we have even the limiting case of a program using system. Recalling our definition of what it is for a system *P* to *use* a program to compute that program, we need a control state s_c such that *P* executes instruction I_1 (which in this case is our only represented instruction) because it was in $f_R(m_1)$ and $f_{\emptyset}(I_1)$, and $f_{\emptyset}(I_1)$ is to have this influence because *P* is in s_c=$C(I_1)$. This requirement is a litte odd because $f_R(m_1)$=$f_{\emptyset}(I_1)$, but the real obstacle is that we do not have the required control state s_c.

So this is where the direct system fails to be equivalent to a program using system. If we had tried to describe the direct system *P* as a program using system that computed the 3-step program $\mathcal{A}$ that the direct system computes, then we would have run into trouble earlier, because we would have needed to provide three distinct physical states in S_A. And then we would have needed three corresponding control states and a much more complicated control mechanism. None of this is required by the direct system.

It is important to emphasize here that we are not denying that any particular physical device that can be described as a direct system might also be a program using system; we are only maintaining that these descriptions would be non-equivalent. It is quite possible that a single physical device could be a direct system that computes one program and also a program using system that computes a program, possibly even the same program, but these would be two systems with different states. And it is important to keep in mind here that we call a system that computes some program $\underline{M}$-π (under some f_r) a direct or hardwired system only if that computation of $\underline{M}$-π is not governed by a control mechanism that operates in the way described in §7.

It is perhaps worthwhile to put these points in another perspective. We have said (in §3.3.) that the memory set M of any machine is a set of n-tuples of symbols. We could have allowed memory sets to include other things, like n-tuples of number, but this may lead to confusion.[4] For one thing, not all computation is numerical. When we calculate we often write symbols down on paper, and we can think of certan computing machines as engaging in a similar activity. To speak of them as "storing" or "manipulating" or "transforming" the references of

----

[4] Clark and Cowell, 1976, for example, use memory sets with n-tuples of numbers.

numerical or alphabetic symbols invites confusion. So we will continue to assume that memory sets contain n-tuples of symbols.

If a memory set of a machine contains pairs of symbols, the machine is a "two-register machine;" if it contains triples, the machine is a "three-register machine," and so on. We may wish to characterize the symbols that can occur in a register of a machine. This might be done by defining a langauge such that the symbols in the n-tuples of the memory set of a machine may be well-formed formulae of that language. We will call such a language a *data language*.[5] in the passages quoted above, Pratt and Burks point out hat the data language of a computing system might well include a programming language; indeed, it might be the very programming language that we use to describe the operation of the system, and the "data" may even include the program that is used by the system to govern its own operation (in those cases where the program treats itself as data). This point is certainly correct.

Notice that is we have a physical system and we can specify a mapping f_R from instructions of some programming language such as LISP, say, to states of the system, this does *not* suffice to show that the system uses LISP as either a programming language or as a data language. We would need to show that the system

[5] Chomsky has pointed out that the operations that a machine can perform on well-formed strings of certain types of data languages provide interesting performance measures for machines. See, e.g., Clark and Cowell, 1976, chs.5-7.

be a computing system at all. If we could do this, or at least provide reason to think that this could be done, then it could be that in those circumstances in which the system computes a program its *data language* is or includes LISP. In order to be justified in saying that the system *used* a progamming language with the LISP syntax, we would need to have some reason to believe that the system was using representations of LISP instructions to control its operation in the way specified above (under some f). And since LISP is an interpreted language, we would not want to say that the system used LISP as a programming language unless in executing any instruction of LISP the system makes an *appropriate* state transition. When we specify a machine *M* that interprets SPL programs, we can say quite precisely what state transitions must be involved in executing an instruction of *M*-SPL under some f_R. An appropriate formal interpretation of LISP would allow us to do the same for LISP instructions.

9. Extending the framework

We chose to develop our account of computing systems around SPL for a number of reasons. Its syntax and semantics are simple and easy to grasp, and it can express any program that we can represent with a flowchart. As Scott points out, "Of course, people want more elaborate programs making use of subroutines and recursive procedures; but still, even those will be compiled into more direct programs that could be illustrated by such flow diagrams (very big diagrams to be sure!)."(1967, p.189) And finally, SPL was ideal for our purposes because its features allow for neat, perspicuous definitions of program computations, and so on. Since we are interested in providing a theory that will have clear application in psychology, neurophysiology and related fields, I think that SPL is all that is needed; there are no serious computational accounts that require any more sophisticated language, but I suppose that this is a controversial claim whose defense goes well beyond the scope of this paper. In any case, it is perhaps worth noting some of the features that we would need to account for to extend our theory to cover programming languages that are more commonly used on our digital computer, if only to give some hint of the territory we have not covered.

All the function and test symbols in SPL are simple primitives, whereas in most common programming languages they may be complex, i.e., built up out of primitives according to various formation rules. This complicates the semantic definition of the language enormously in most cases. The operation and test symbols of most languages have mneumonic content to aid the programmer, but often this content gives but little indication of what an appropriate formal interpretation of them would be. Consider, for example, "assignments," which are used very frequently in programming,

X := X + Z,

X ← X + Z, or

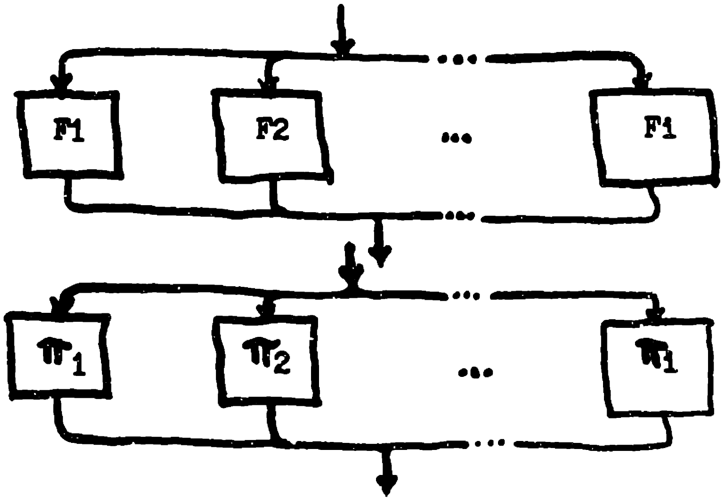
X = X + Z,

all of which would be read, "Set X equal to X plus Z." In all of our examples, the interpretation of an operation symbol has been a mapping over n-tuples of numerals, but in most languages we will need not only numerals but also other "data structures" of various "types." Somehow these must be used to interpret assignments of values to programmer defined names. There are "declarations" which associate names with data-types and "type-checking operations." Specifying appropriate interpretations for such operation symbols is no trivial problem; the best way to handle them is still a matter of controversy. (Cf., e.g., Stoy, 1977, and Brady, 1977.)

Various kinds of sequencing operations that we do not have in SPL are also included in many programming languages. In SPL a program is just a set of labelled instructions, each of which (with the exception of halt instructions, of course) specifies the label of the next instruction to be executed. In most programming languages, however, the programs are ordered sets of instructions, and the order in which operations are performed is typically specified partly by this ordering, and partly with parentheses and precedence rules, and sometimes by the instructions themselves as in the case of a "jump" or "subroutine call."

Some languages also allow the programmer to write instructions that call for the concurrent execution of a number of operations or subroutines. We could incorporate this capability in an extended SPL also. We could, for example, allow structures like (FLOWCHART $\mathcal{P}$) in our flowcharts. And the

(FLOWCHART $\mathcal{P}$)



definition of a computation could be modified analogously to allow for this concurrency. Programs that call for this sort of parallel processing are usually simulated on a conventional computer that uses sequential processing (or something very much like it), but this is for practical reasons and not for reasons of principle. Computers that make heavy use of parallel processing and programming languages for them are being developed. (Cf., e.g., Dennis and Misunas, 1974, and the references cited there.)

In general, then, computing a program of any programming language involves passing through an appropriate series of memory states in any computation, as it did in SPL languages. The specification of the appropriate memory states will typically be much more complicated than it was for computations of SPL programs, however, and concurrent processing will require some special treatment. Since it looks like computing a program can generally be described in such a way, the framework adopted here looks like it will allow a natural extension to the other programming languages. The major change will be in *how* the sequence of memory states that occurs in any computation will be specified. Defining a computation will typically be much more difficult.

10. Highlights for computational psychology

In this chapter we have provided relatively clear definitions of a number of terms that are commonly used in computationaal psychology. We will tenatively accept these definitions and attempt to explicate and assess some of the psychological claims in which these terms occur in the next chapter. But we can note here some important points that we get from our theory immediately.

10.1. Three grades of program involvement

It is worth emphasizing again the three grades of program involvement that have been pointed out already. In a computational theory we may use some program (e.g., an *M*-program π) to define a partial function ($\underline{M}_{\pi}$) that is computed by a system; we will call this the "first grade of program involvement." This requires interpreting the system under some f_R such that in certain circumstances the system will always proceed from some "initial state" ($f_R(m)$) to the corresponding "final state" ($f_R(\underline{M}_{\pi}(m))$). A theory that is committed only to this first grade of program involvement we will call a "first level theory."

The transition from an initial state to a final state may be done in a number of steps; we may be able to show that the transition is accomplished by executing some program with more than one step. The specification of a program (with more than one step) that is *computed* by the system under some f_R, then, is the "second grade of program involvement."

A system that computes any program may do so either by being "hardwired" to do so, or by using a representation of the program to control its operation; any program that can be computed one way can be computed the other way also. And we could have a "hybrid" system that uses program using systems only to compute some of the steps of the programs it computes. The specification of a program *used* by a program system to control its operation is the "third grade of program involvement." A theory that is committed to the existence of a program using system is a "third level theory."

Whereas the first two grades of program involvement use the program to describe what the system does under some f_R, we ascend to the third grade only when we are accepting a commitment to a certain account of *how* the system computes the programs it computes. The third grade of program involvement commits us to the existence of some "control mechanisms" that govern the operation of the system according to a representation of the program computed. We can distinguish these three grades of program involvement regardless of whether the programs are written in SPL or some other programming language.

10.2. Formality constraints

One constraint on computational descriptions of physical system has been called the "formality constraint" by Fodor (1980). Computational processes are, he says, "symbolic and formal:" "They are symbolic because they are defined over representations, and they are formal because they apply to representations in virtue of (roughly) the syntax of the representations."(p.64) If we use the word 'representation' rather loosely so that it applies to any memory states in any set $f_{\pi}[r(\pi)]$ or in any n-tuple in any set $f_R[M]$, then Fodor's claim follows trivially from our definitions of program computation and program use. If s is some state in a set $f_R[M]$, for example, then our definition of program computation by a physical system guarantees that s will influence the computational processes of the system according to which symbol in M it corresponds to rather than according to what it means. The state transitions the systems makes in the execution of any program depend on which memory states the system is in, not on what those memory states refer to or represent. We can safely describe a computation in terms of what the memory states represent (as we did in §3.5, for example), only if distinct memory states represent distinct objects. And of course, the formality constraint applies in the same way to both data languages and programming languaegs; in either case, the operation of the system is specified according to which memory state (i.e., which state of $f_R[M]$ or $f_{\pi}[r(\pi)]$) the system is in.

We could modify our theory of computation so that one memory state would have different causal roles depending on what it represented. We could have, for example, a programming language with an ambiguous instruction, i, such that i could be properly executed by performing either of two different state transitions, and so $f_{\pi}(i)$ could have either of two different causal roles in a system using this language. But there would be no point to giving such a description of a system in most cases. If $f_{\pi}(i)$ has one causal role when the system is in state s and otherwise has the other causal role, then the system is better described as one that uses a language with two unambiguous instructions, i' and i'', such that $f_{\pi}(i')$ is the state of being in $f_{\pi}(i)$ and s, and $f_{\pi}(i'')$ is the state of being in $f_{\pi}(i)$ but not in s. It is hard to see why a non-deterministic description would ever be preferred if a deterministic one were available. But this sort of indeterminacy is, in effect, what we would allow if we allowed the causal roles of our representations to depend on semantic properties of the representations, since then the state transitions at respective points of the computation would not in general be completely determined by the system's formally defined memory states.

BIBLIOGRAPHY FOR CHAPTER III

Arbib, M.A. (1972) Toward an automata theory of brains. *Communications of the ACM*, *15*, pp.521-527.

Brady, J.M. (1977) *The Theory of Computer Science: A Programming Approach*. New York: Wiley.

Brainerd, W.S. and Landweber, L.H. (1974) *Theory of Computation* New York: Wiley.

Burks, A.W. (1963) Programming and the theory of automata. In P. Braffort and D. Hirschberg, eds., *Computer Programming and Formal Systems*. Amsterdam: North-Holland.

Clark, K. and Cowell, D. (1976) *Programs, Machines, and Computation: an Introduction to the Theory of Computing*. New York: McGraw-Hill.

Cummins, R. (1977) Programs in the explanation of behavior. *Philosophy of Science*, *44*, pp.269-287.

Dennis, J.B. and Misunas, D.P. (1974) A preliminary architecture for a basic data-flow processor. Project MAC, Computation Structures Group Memo 102, MIT.

Dennis, J.B. (1976) A language design for structured concurrency. In G. Goos and J. Hartmanis, eds., *Design and Implementation of Programming Languages*. New York: Springer-Verlag.

Dertouzos, M.L., Spann, R.N., Athans, M., and Mason, S.J. (1972) *Systems, Networks, and Computation; Basic Concepts*. New York: McGraw-Hill.

Dijkstra, E.W. (1972) Notes on structured programming. In D. Dahl and C. Hoare, eds., *Structured Programming*. New York: Academic Press.

Dijkstra, E.W. (1976) *A Discipline of Programming*. Englewood Cliffs, New Jersey: Prentice-Hall.

Feldman, J.A. (1979) Programming languages. *Scientific American*, *241*, pp.94-116.

Fodor, J.A. (1980) Methodological solipsism as a research strategy in cognitive psychology. *The Behavioral and Brain Sciences*. *3*, pp.63-109.

Flynn, M.J. (1975) Intepretation, microprogramming and the control of a computer. In H.S. Stone, ed., *Introduction to Computer Architecture*. Chicago: Science Research Associates.

Haugeland, J. (1978) The nature and plausibility of cognitivism. *The Behavioral and Brain Sciences*, *2*, pp.215-260.

Newell, A. and Simon, H.A. (1976) Computer science as empirical inquiry: symbols and search. *Communications of the ACM*, *19*, pp.113-126.

Pratt, T.W. (1975) *Programming Languages: Design and Implementation*. Englewood Cliffs, New Jersey: Prentice-Hall.

Pylyshyn, Z. (1980) Computation and cognition: issues in the foundation of cognitive science. *The Behavioral and Brain Sciences*, *3*, pp.11-169.

Scott, D.S. (1967) Some definitional suggestions for automata theory. *Journal of Computer and System Sciences*, *1*, pp.187-212.

Scott, D.S. (1977) Logic and programming languages. *Communications of the ACM*, *20*, pp.634-641.

Stoy, J.E. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*. Cambridge, Massachusetts: MIT Press.

Tennant, R.D.(19760 The denotational semantics of programming languages. *Communications of the ACM*, *19*, PP.437-453.

Von Neumann, J. (1958) *The Computer and the Brain*. New Haven: Yale University Press.

Von Neumann, J. (1966) Theory of Self-Reproducing Automata. Edited and completed by A.W. Burks. Urbana: University of Illinois Press.

CHAPTER IV

PROGRAMS AND RULE GOVERNED BEHAVIOR

> "...take Newton's law for gravitation...it is kind of mathematical, and we wonder how this can be a fundamental law. What does the planet do? Does it look at the sun, see how far away it is, and decide to calculate on its internal adding machine the inverse of the square of the distance, which tells it how much to move? This is certainly no explanation of the machinery of gravitation! You might look further, and various people have tried to look further. Newton was originally asked about his theory - 'But it doesn't mean anything - it doesn't tell us anything'. He said, 'It tells you how it moves. That should be enough. I have told you how it moves, not why.' But people are often unsatisfied without a mechanism..."
>
> -- Richard Feynman

1. This chapter will consider whether there is evidence to support claims to the effect that some human behavior is "rule governed" in something like the way that the behavior of a programmed computer is "rule governed," claims to the effect that in certain cases the behavior of an organism is the result of a computational process that is governed by a program, a set of internally represented rules or formulae of some programming language. There have been many proposals that might be construed in this way. Von Neumann(1958) speculated that the brain uses a higher level programming language (or, in his terms, a "short code") of a peculiar kind, and this sort of claim has since become quite popular. J.Z. Young(1964) proposed that this computational account was the appropriate one for

neurophysiologists, and he has been followed in this by Arbib and others (see Szentagothai and Arbib, 1975). Similar accounts have been adopted by psychologists. Motor learning theorists talk about the programming of motor behavior (as in the papers in Stelmach, 1978), and psycholinguists compare natural languages to higher level programming languages which are compiled and then executed (Fodor et al., 1974; Johnson-Laird, 1977). But we will concentrate on the claim made by some linguists that linguistic behavior is somehow governed by an internalized grammar. This is no doubt the claim that has received the most attention, and the computer analogy has frequently been appealed to in the controversy it has given rise to.

2. Our discussion will presuppose some understanding of computation. All the necessary background was covered in some detail in the last chapter, but we will briefly review the leading ideas.

Despite the difficulties we may have in grasping or spelling out the details of any particular computational description of a physical system, there should not be anything mysterious about computational description in general. This is, after all, just a certain way of describing the operation of physical systems, a way that is often quite clear and particularly useful. What we call "calculators" and "computers" are basically just physical systems which are, by design, conveniently described in this way and useful for this reason. But actually, any physical system can be given <u>some</u> computational description or other. This can be done by specifiying a (1-1) mapping f_R (a "realization" function) from physical states of the system into some set of symbols, so that changes of physical state correspond to symbolic transformations. When these changes in physical state are regular and predictable, so are the associated symbolic transformations. So, for example, sometimes we can specify a mapping f_R such that in some specifiable circumstances C the system will always, in virtue of physical laws that apply to the system, go from one state s_i into another state s_f, where for every such pair of states, the symbol $f_{R(s_f)}$ is a function F of of the symbol $f_{R(s_i)}$. This is the basic basic idea upon which all computational descriptions rest; we shall we shall call this a "first level" computational description.

Since programs are so common in computational descriptions, it is perhaps worthwhile to note that a program may be used in providing even a first level description. For example, if the function F computed by a system (under some interpretation f_R, in some circumstances C) is quite complicated, we may find it useful to provide an algorithm for computing its values, and this algorithm may be expressed in a programming language. A program that yields a unique value for each different input defines a (partial) function. So a program may be used even in a first level theory to specify the (partial) function computed by the system.

Programs sometimes provide more than just a specification of the function computed. A program typically expresses an algorithm according to which the input symbols are transformed into other symbols in a sequence of steps. The symbols that result from the execution of every step in the procedure may be in the range of the interpretation f_R, in which case the system may compute the function specified by passing through each of the intermediate states; it may execute each appropriate step in the program by computing the function specified by that step. In such a case we will say that the system *executes* or *computes the program*; it computes the function specified by the program by computing the program in this way. This is the "second level" of program involvement.

Programs may have even more of a role in a computational theory than this. A physical system may compute a program because its computation of that program is governed by a representation of that program. We will call such a system a "program using" system. In this "third level" of program involvement, the appropriate steps of the program are executed because a control mechanism brings it about that the representation of the appropriate instruction (specified by a "program realization" function f) determines the change of state at each point; the system *uses the program* to govern its operation.

As was pointed out in the last chapter, any program that can be computed by a program using system can be computed directly by a system that we would clearly not want to call a program using system. The Mars-Sun orbital system computes a program under some interpretations of its states, for example. In the last chapter we showed that it computes a program which computes solutions to Newton's law of universal gravitation. But it is clearly not a program using system. And electronic circuits can be "hardwired" to compute even very complex programs directly, without having control mechanisms to govern their operation according to a representation of the program. Any program that can be computed by any system can be computed directly, or by a program using system, or by a "hybrid" system that uses a program to govern its computation of only certain steps of the program.

Notice that there is an important difference between first and second level computational descriptions, on the one hand, and third level computational descriptions, on the other. All three grades involve giving a symbolic interpretation of the relevant states of the systems, but whereas the first two grades of program involvement use the program to describe how the symbols are transformed or "manipulated," we ascend to the third grade only when we are also making a claim about how the specified computations are carried out. The third grade commits us to the existence of some mechanism that controls the computational processes of the system according to a representation of the program. Third level theories are those that maintain that the computation is carried out by a program using system, not by a direct or hybrid system.

3. The primary aim of this paper is to explore the methodology for confirming "third level" computational theories of human psychological processes. As was noted above, we will focus on a particular theory that has received wide attention, viz., the theory that language users employ an "internalized" grammar when they exercise their linguistic abilities. Do proponents of this theory mean to suggest that the grammar is represented and utilized by a program using system? As we will see, some of their remarks suggest that they do, but we need to be careful here because we are bringing distinctions to bear that are not usually recognized.

The theory I want to consider has been most clearly formulated by Noam Chomsky. He says,

> To know a language, I am assuming, is to be in a certain mental state, which persists as a relatively stable component of transitory mental states. What kind of mental state? I assume further that to be in such a mental state is to have a certain mental structure consisting of a system of rules and principles that generate and relate mental representations of various types. (1980b, p.5)

This "system of rules and principles" is the grammar of the language, so the following hypothesis is apparently being proposed:

(H1) The grammar is "mentally represented" and used in language processing.

Chomsky urges that in proposing this view he is claiming that language behavior is "rule-governed" behavior; in exercising our linguistic abilities we are "following" the rules of the grammar rather than merely conforming to them.(1980b, pp.13,54-55) The evidence for this view is that it explains certain facts about our language and that no better theory is available:

> The evidence bearing on the hypothesis attributing rules of grammar to the mind is that sample facts are explained on the assumption that the postulated rules are part of the AS [the "attained state" of the language learner] and are used in computations eventuating in such behavior as judgements about form and meaning. (1980b, p.54)

> I know of no other account that even attempts to deal with the fact that our judgements and behavior accord with and are in part explained by certain rule systems (or to be more accurate, are explained by theories that attribute mental representations of rule systems)...The critic's task is to show some fundamental flaw in principle or defect in execution, or to provide a different and preferable account of how it is that what speakers do is in accordance with certain rules -- an account that does not attribute to them a system of rules (rules which in fact appear to be beyond the level of consciousness). (1980b, p.12)

In short, we need to propose the represented rule system to explain certain facts about language processing. Language processing is distinguished in this respect from simpler processes that presumably do not involve rules. Things like a person's riding a bicycle(1969, pp.154-155), the flight of a bird (1975, pp.222-223), or the flight of a pigeon-controlled missile (1980b, pp.10-11) can presumably be explained in terms of "reflexes" or other relatively simple mechanisms.

There are three main points in this sort of position that suggest that a "third level" theory is being proposed. First, it is claimed that certain rules are represented. Second, these rules are assumed to govern language processing; they are used to "generate" mental representations that are used in language understanding, for example. And finally, this sort of system, a system that uses represented rules, is distinguished from simpler systems that do not use them. These three points are precisely what a third level theory would be committed to. It would have a commitment to represented rules or instructions which control the computation. And whereas virtually any system can be given a "second level" computational description according to which representations are transformed, a third level computational description according to which the system computes a nontrivial program is true only of certain quite complex systems. So let's make a third level construal of these claims explicit.

The grammar itself is not a language processing algorithm, of course. We could assume only that the grammar is <u>used</u> in the course of executing such an algorithm. The claim that the grammar governs language processing, then, might be construed as entailing the following sort of hypothesis with respect to each linguistic ability,

> (H2) Language understanding involves the computation of some program P, a program that includes the rules of the grammar, G, which are executed to generate the mental representations needed for the computation.

And so then (H1), the claim that the grammar is "mentally represented" and used, would be construed as committing us to the hypothesis,

> (H3) In human language understanding, the computation of P is carried out by a program using system whose operation is controlled by a representation of P and hence also of G.

This proposal is clearly a third level theory.

I will argue that this third level construal of (H1) is untenable. But it is important to keep in mind that we will attempt to consider only the issues relevant to this explication of the linguists' claim. One might challenge both (H1) and (H3) on the ground that there is another theory that does not presume that the grammar is involved in the linguistic processes at all; one might argue, for example, that speakers use some sort of "heuristics" in language understanding, or that they compute a recognition algorithm derivable from a lexical-interpretive grammar. (Cf., e.g., Fodor et al., 1974, ch.6, and Kaplan and Bresnan, in press.) All of these theories can be construed as making different second level claims about what procedures are computed in language understanding, and with respect to each of them we could ask whether there is evidence that the postulated procedure is represented and in control of the computation. But (H1) is the best candidate for such consideration even if it is not the best theory on empirical grounds because, in the first place, the issues in every case will be analogous to those encountered in this attempt to explicate (H1), and in the second place, only discussions of (H1) have given any particular attention to those issues that are relevant to this project. Thus, although a similar consideration could be given to any of the competing theories, an attempt has been made here to avoid getting tangled in any problems that are not essential to making a few restricted points about third level and other computational explications of (H1); we will not consider whether (H1) should be rejected in favor of one of the competing views.

In §4 it is argued that evidence typically advanced in support of (H1) does not support (H3). In §5 other evidence is considered which is, I think, offerred specifically in support of (H3) but which fails to provide any substantial support. And in §6 I note that there are good reasons for supposing that Chomsky and other linguists have <u>not</u> intended (H1) to be construed as (H3), and other possible construals of (H1) are briefly considered.

4. In this section it will be argued that the evidence that linguists offer in support of (H1) does not support (H3). The evidence for (H1) comes from three related sources in linguistic theory:

(1) the explanation of language comprehension and other linguistic abilities;

(2) evidence for formal properties of grammatical rules, and

(3) the explanation of language acquistion.

Each of these sorts of evidence will be considered in turn with regard to whether it provides substantial support for (H3).

4.1. Chomsky gives some examples of the sort of evidence that is used to support (H1) in a recent paper (Chomsky, 1980b). Linguists discover some regular relation between declarative sentences and the corresponding interrogatives, or some constraints on anaphoric relations between a reciprocal expression and its antecedent, and so they propose that

> some general principle of language applies to permit the proper choice of antecedent -- not an entirely trivial matter...Similarly, some general principle of language determines which phrases can be questioned.(p.4)

The proposed set of rules and prhnciples, the grammar, is assumed to be "one basic element in what is loosely called 'knowledge of language'." Chomsky says,

> I am assuming grammatical competence to be a system of rules that generate and relate certain mental representations, including, in particular, representations of form and meaning, the exact character of which is to be discovered, though a fair amount is known about them. These rules, furthermore, operate in accordance with certain general principles. I have informally discussed rules of grammar that, for example, move a question-word to the front of a sentence or associate an antecedent with an anaphoric expression such as "each other."...the movement rule is governed by a principle of "locality" -- elements of mental representations can't be moved "too far" -- and the choice of antecedent is governed by a principle of "opacity" -- variable-like elements can't be free in certain opaque domains, in a sense specific to the language faculty. (p.10)

To explain linguistic abilities, such as the ability to understand language, we assume that the grammar is <u>used</u>. This explains why the speaker of the language respects the generalizations captured by the grammar in his exercise of these abilities. That is, this explains why the speaker's performance respects the grammar insofar as it does; presumably other performance factors will be required to explain why the speaker's performance does not always respect the grammar. In any case, we have an argument for supposing that the grammar is mentally represented and used in language processing, as (H1) claims.

Granting that this is one of the basic forms of argument for (H1), our question is whether it has any plausibility as an argument for (H3). If this argument does provide support for such a computational account, then we should be able to render it in a form that is explicit about its computational commitments. So, to begin with, there is perhaps a tenable argument here to the effect that grammatical rules are computed in language understanding, that the grammar G is somehow "embedded" in whatever procedure is computed. One might argue as follows: The only plausible explanation of how a person understands a sentence is that he formulates the various representations of the sentence that are generated by the grammar, the representations over which the non-generative rules and principles are defined. The most plausible account of this process is that the grammar itself is employed in the computation, its rules being "followed" or "executed" to <u>generate</u> the needed representations at the appropriate points of the computation and to <u>apply</u> the other appropriate rules and principles to the representations so generated. Roughly, the proposal is that the speaker cannot understand the sentence unless he recognizes its words and syntax, and he cannot recognize its words and syntax unless he recognizes its phonetic structure, and so on. An analysis-by-synthesis algorithm, examples of which will be mentioned below, is a procedure of this kind, though it is perhaps only a crude first guess at what might be going on. There may be other procedures that make much more efficient use of the grammatical rules. The argument that some such procedure is computed, though, is just that it is plausible that a procedure that has G embedded in it could presumably be one whose computation would respect the generalizations captured by the grammar. So this supposition about the procedures computed could explain why many diverse aspects of the speaker's language processing accord with the grammar -- the procedures computed actually involve the execution or application of grammatical rules.

This is an argument for (H2), though, and not an argument for (H3). That is, this argument supports the proposed view about what program is computed in language understanding, but it does not support any particular view about what mechanisms are responsible for this computation. Since any computable program can be computed by a "direct" or "hardwired" system, or by a "hybrid" system, or by a "program using" system, the attribution of any of these mechanisms would suffice to explain why the system computes the program it computes; any of these explanations can account for "how it is that what the speakers do is in accordance with certain rules, or is described by these rules." Thus, further evidence is needed to support any particular claim about which sort of mechanism is, in fact, responsible for the computation. So even if we assume that this argument for (H2) is a good one, the question is: What evidence supports the further claim that the program P and hence also the grammar G are mentally represented and controlling the computation? I am unable to see any grounds in the sort of argument presented here for the claim that the rules of grammar are not only computed to generate the needed representations, but also represented and followed in language processing. Although it is not clear that he would accept our account of computation, I think that exactly the right point is made for us by John Searle(1980) in the following passage:

> The claim that the agent is acting on rules involves more than simply the claim that the rules describe his behavior and predict future behavior. Additional evidence is required to show that they are rules the agent is actually following, and not mere hypotheses or generalizations that correctly describe his behavior; there must be some independent reason for supposing that the rules are functioning causally. (p.37)

This is precisely the right point to be made about the difference between (H2) and (H3): (H2) asserts only that the language processing conforms to the rules of the grammar, while (H3) asserts that the processing conforms to the rules because the rules are represented and followed.[1]

If Chomsky is proposing (H3), then we have done what he says the critic must do: "to show some fundamental flaw in principle or defect in execution, or to provide a different and preferable

[1] I suspect that Searle would want to criticize not only the third-level theory that the rules are represented and used but also the second-level theory that they are computed. However, in this chapter we are considering only the former claim. An examination of the methodology for confirming second-level theories like (H2) will have to be left to another paper.

account of how it is that what the speakers do in in accordance with certain rules -- an account which does not attribute to them a system of rules..."(1980b, P.12) The fundamental flaw in principle is the failure to recognize that additional evidence is needed to support the claim that the rules are not only computed but also represented and used. The alternative account that does not attribute represented rules is the theory that the rules are computed "directly" or by a "hybrid" system, by some system that is not a program using system, and we have seen in the last chapter that there will always be possible systems of this sort. This alternative may, in fact, be preferable in the absence of any evidence against it, since direct, hardwired computation is typically faster and more efficient than computation on a program using system. If, on the other hand, Chomsky and other linguists do _not_ mean to propose (H3), then the failure to provide the needed evidence is not in the least surprising.

4.2. Another line of argument for (H1) that we ought to consider has been pointed out by Fodor, Bever and Garrett (1974) and others. They point out that linguists often seem to be making claims about formal properties of the rules of the grammar, and only representations have formal properties. If hypotheses about formal properties of the rules are needed to explain certain data, this surely supports the view that the rules are represented. So we ought to consider whether there is any such evidence relevant to the formal properties of the rules, and if there is, whether _that_ evidence supports (H3).

Fodor, Bever and Garrett(1974) point out this line of argument in a discussion of how linguistic universals ought to be accounted for:

> ...there are linguistic universals which serve precisely to constrain the form in which information is represented in grammars (i.e., the form of grammatical rules). The question is: If the universals do not also constrain the form in which linguistic information is represented in a sentence-processing system, how is their existence to be explained? Surely if universals are true of anything, it must be of some psychologically real representation of the language. But what could such a representation be if it is not part of a sentence encoding-decoding system? (pp.369-370)

I think that Fodor et al. have fallen prey to a confusion here about the status of linguistic universals which are stated as constraints on the form of grammatical rules. In fact, most linguistic universals do _not_ involve any commitment to rules of a certain form; rather, they involve formally specifiable constraints on the applicability or operation of the rules. Thus, we can think of them as generalizations that are true of the _operations_ performed on the linguistic structures posited by the theory. This distinction is crucial to the present point, but it is no surprise that it is occasionally overlooked in the linguistic literature where it is usually insignificant.

Chomsky makes the relevant point in the following passage from "Conditions on Transformations"(1973). He says,

> For heuristic purposes we may distinguish two aspects of universal grammar: (a) conditions on form, and (b) conditions on function -- that is, (a) conditions on systems that qualify as grammars, and (b) conditions on the way the rules of the grammar apply to generate structural descriptions. (p.232)

It is the "conditions on form" that Fodor et al. apparently have in mind, but Chomsky rightly emphasizes that the distinction between these and "conditions on function" is made only "for heuristic purposes:"

> The distinction is one of convenience, not principle, in the sense that we might choose to deal with particular phenomena under one or the other category of conditions. (p.232)

This point is really quite clear. Suppose that we have one set of rules, call them "root transformations," which must apply after another set of rules called "cyclic transformations." If it is a universal property of grammars of all possible human languages that there be two such sets of rules whose application must be ordered in this way, we could capture this fact by beginning all and only the cyclic transformations with the dummy symbol 'C' and proposing:

(1) All root transformations must apply after rules beginning with 'C'.

To capture the universal in this way we <u>do</u> need to require that certain rules have a certain form (viz., that the cyclic rules begin with 'C'), but there is, of course, no need to express the universal in this way. Instead, we could just distinguish root transformations from cyclic transformations on the basis of their operation and say,

(2) Root transformations apply after cyclic transformations.

This latter claim does not commit us to rules having any particular form, but only to a constraint on the order in which certain operations can be applied to linguistic representations.

Perhaps this point should be illustrated with a more likely example. Let's consider one of Chomsky's examples of a "condition on form:"

> ...consider the definition of a transformation as a structure-dependent mapping of phrase markers into phrase markers that is independent of the grammatical relations or meanings expressed in these grammatical relations. This definition makes certain operations available as potential transformations, excluding others...By requiring that all transformations be structure-dependent in this specific sense, we limit the class of possible grammars, excluding many imaginable systems. (1973, p.233)

Ironically, this universal that is offerred as a "condition on form" has here been expressed informally as a "condition on function;" grammatical operations apply to linguistic representations having a certain structure, not only to representations having a certain meaning or a certain number of words, for example. In fact, Chomsky never does express this formally as a "condition on form," but his discussion of it indicates that what he has in mind is that we will capture the structure-dependence of the rules by expressing each one in such a way as to indicate its dependence on structure. He mentions the passive transformation, for example; any of the various typical ways of writing this rule will indicate that it applies to phrase markers with a certain structure, as in,

(3) NP_1, Aux, V_x, NP_2 => NP_2, Aux, BE, by + NP_1.

In this case, the form of the rule (under its standard interpretation) indicates to the linguist when it can apply and what it does; for example, the symbols to the left of the arrow indicate that the rule applies to strings that can be factored into four successive substrings, the first and last of which are noun phrases, the second an auxiliary, and the third a verb of a particular category. It is, no doubt, convenient to express our grammatical rules in some such form, but the structure-dependence of the grammatical operations does not require us to do so. The structure-dependence of the rules is, in fact, entirely neutral with regard to the formal properties of the rules.

4.3. It is at least not obvious that there are any supportable claims that really commit us to rules of a certain form, though there is clearly a general pressure in linguistics to constrain the formal properties of rules. This pressure comes with the acceptance of a certain approach to what Chomsky has called "the central problem of linguistic theory," the problem of explaining how a child can master a language given only limited and degenerate evidence. Chomsky has proposed that the way to solve this problem is to assume that language acquisition involves testing hypotheses; the child selects the grammar of his language from the class of possible grammars on the basis of linguistic evidence:

> The child is presented with data, and he must inspect hypotheses (grammars) of a fairly restricted class to determine compatibility with this data. Having selected a grammar of the predetermined class, he will then have a command of the language generated by this grammar. (Chomsky, 1972, p.159)

This proposal assumes, then, that the possible grammars considered in this selection process are represented, and to make the learning task managable it is essential that the set of possible grammars that need consideration be severely constrained. Thus Chomsky says, "Reduction of the class of possible grammars is the major goal of linguistic theory."(1977, p.125) If the grammars are represented, then the grammatical rules must have some formal properties or other, and so there is a pressure to discover what they are.

Since the problem is to constrain the class of possible grammars that need to be considered by the language learner, linguists have adopted the reasonable strategy of assuming that whenever we discover that grammars of natural languages can be constrained in a certain way, we should assume that they are constrained in that way. This is where linguistic universals have an important role to play. But we should distinguish here the various kinds of constraints that would be relevant to this "central problem of linguistic theory."

We would like, first, constraints on what Culicover et al.(1977, p.1) call "the expressive power of the descriptive devices used in writing grammars." That is, we would like to constrain the set of languages that any rules of grammar could generate. Notice that constraints of this sort are not constraints on the form of the rules, but constraints on what they can do, constraints on their formal generative power. We would also like to impose "constraints on functioning" that have the effect of limiting the applicability of certain kinds of rules (or of certain particular rules). Let's follow Chomsky and continue to call constraints of this second kind "constraints on functioning." As Chomsky and others have pointed out, these constraints are potentially just as valuable as constraints on generative power since they will serve to limit the class of grammars that need to be considered given some body of evidence; that is, they will allow the language learner to rule out grammars that, *with* the constraints on functioning, cannot accomodate the linguistic evidence. And finally, a third class of constraints that we would like to have is a class of constraints on the formal properties of the represented grammatical rules; *these* are properly called "constraints on form." If the rules *are* represented, they must have formal properties, and we would like to discover what they are. It is this last set of constraints that concern us here.

How should the formal properties of the represented rules be constrained? This question has not received proper attention because it is not carefully distinguished from questions about the other kinds of constraints. Even the most basic formal properties remain essentially uninvestigated. But this is not too surprising. For one thing, the general pressure to constrain the class of possible grammars that the child needs to consider provides the linguist with a natural strategy with respect to the other two kinds of constraints, viz., that whenever we can constrain the class of grammars in a certain way, we should. But the problem is more difficult in the case of constraints on the formal properties of the rules.

We can suppose that no two grammars differ *only* in formal properties, that there are no two formally distinct grammars that generate exactly the same linguistic structures in exactly the same way. We can impose this constraint without losing explanatory power, so we will. Thus we assume that we are looking for a system that has at most one way of expressing any particular rule. But what can we say about the formal properties of the rules? It was noted above that certain linguistic universals *could* be captured by constraining the form of the rules in certain ways (given some standard interpretation of the formalism). Thus, if the only quantifiers in the system are '∀x' and '∃x", and we think that transformational rules do not need quantifiers, the we can propose that no rules contain '∀x' or '∃x'. But clearly the point of the constraint is to rule out *any* symbol that is *interpreted* as a quantifier; that is, the constraint is not really a formal constraint at all. And we could assume that the rules contain symbols like 'NP', 'VP' and so on, but as we noted above, the point of this assumption is to rule out operations that are not structure dependent.

In all of these cases we have constraints that will bear on what is expressed in the formalism of the represented rules, but we have no indication as to the properties of the formalism itself. No one has confronted even the most basic questions. Why should we assume even that the representations of the rules are complex (i.e., built up from primitive elements)? Halle and Keyser(1971, p.8) propose that, all other things being equal, we should prefer rules containing the least number of symbols. So why couldn't we, for example, express each rule with a decimal numeral (as we might express the instructions of a simple machine language for a computer)? Perhaps one could give reasons for thinking that this would not do, but the linguistic literature does not address this sort of question. Rather, (as in Halle and Keyser, 1971, pp.4-5) some system of notation that is perspicuous is just adopted, and then constraints on generative power and on functioning are sometimes expressed in this notation under its standard interpretation.

In sum, the hypothesis testing theory of language acquisition has not, I think, provided any particular formal constraints on grammatical rules. But the fact that this approach presupposes that the grammar is represented itself gives support to (H1), unless there are other plausible ways of handling the central problem of linguistic theory. The hypothesis testing model has, in fact, been criticised (see, e.g., Braine, 1971), and alternative theories have been proposed. In recent work Chomsky has modified his account. He says in a recent paper,

> I will assume that universal grammar provides a highly restricted system of "core grammar," which represents in effect the "unmarked case." Fixing the parameters of core grammar and adding more marked constructions that make use of richer descriptive resources, the language-learner develops a full grammar representing grammatical competence. (1980c, p.3)

It is not quite so obvious that this sort of proposal requires that either the core grammar or the acquired grammar be represented, but such representation is apparently still assumed. Of course, this new proposal is not completely worked out, and it faces some difficult problems. (See, e.g., Pinker, in press.) But there is little doubt that Chomsky's basic insight is correct: to account for language learning we must postulate a "rich internal structure;" we must assume that the language learner comes to the learning situation already well equipped so that he needs only very limited linguistic experience to master the language spoken in his community. The search for linguistic universals is clearly relevant to discovering what the language learner might bring to the learning situation, and hence relevant to any such theory of language acquisition, whether the grammar is represented or not.

Now we want to consider whether the explanation of language acquisition provides any grounds for (H3), so again let's consider the mentioned views with explicit attention to their computational commitments. Suppose that language acquisition does involve representing grammars and testing them against the evidence. This is apparently a second level claim about what sort of computation goes on in the process of language learning. But since selecting the right grammar in this process is an essential part of what gives the language learner mastery of the language generated by the grammar, it *is* plausible that this grammar is *used* in the exercise of linguistic abilities as (H3) claims. That is, if we assume that the appropriate grammatical rules are computed in the course of language understanding *because* the language learner has selected the right grammar, then (H3) seems like the best account. But the hypothesis testing theory of language acquisition has serious problems, and alternative theories have been proposed, as was noted above. So we need to consider whether these alternatives also give support to (H3).

The proposed alternatives are like the hypothesis testing theory in assuming that the language learner comes to the learning situation specifically equipped to learn a language with a certain special, intricate structure, i.e., a language with the structure that is characteristic of human languages. However, as was observed above, it is not at all obvious that these alternatives require that we assume that the grammar is mentally represented. Similarly, it appears that they do not support (H3). The parameter-setting proposal is, I think, particularly amenable to the view that the grammar is not represented. We might assume, for example, that there is a core grammar which is not represented and in control of the processing but essentially "wired in" and that only certain adjustments and additions need to be made in order to yield a full linguistic competence.

The view that the core grammar might be built into the system that executes linguistic procedures directly comports well, I think, with Jerry Fodor's proposal that language recognition processes and low level sensory processes are "modular."[2] That is, they are fast "bottom-to-top" processes that make use of only a restricted domain of information. These modular processes all develop in characteristic ways in the course of normal human development, and each seems to be subserved by particular neural structures. Such processes are

[2] Jerry Fodor presented this thesis in a series of lectures given at MIT in the fall of 1980.

perhaps the ones in which computation is most plausibly carried out directly, without the mediation of a control mechanism that must access a representation of the procedure to be executed.

5. So far it has been argued that evidence typically offerred in support of (H1) does not support (H3): (H3) is not needed to explain the fact that the generalizations captured by the grammar are respected in the exercise of our linguistic abilities; neither is it supported by various claims that are apparently about formal properties of grammatical rules, and it is not required for the explanation of language acquisition. In this section other sorts of evidence for (H3) will be considered, evidence that has apparently been offerred specifically in support of hypotheses like (H3). Such hypotheses have been discussed primarily in the literature concerned with speech recognition models.

5.1. As has been emphasized already, to support (H3) we must do more than just support a claim like (H2) about what procedure is computed, since any procedure can be computed in any number of ways that do not require any representation of the procedure. It is not obvious what sorts of evidence would support the additional claim of <u>any</u> third level theory. Given <u>any</u> account of what program is computed, what evidence would support the additional claim that this computation is controlled by a representation of the program? If we can find any good method for confirming third level theories, then it will be of interest to consider whether that method provides any support for (H3).

It would not be expected that there would be well developed methods for confirming third level theories unless the claims of third level theories were at least distinguished from those of second level theories. So perhaps it is worthwhile to digress here to consider what others have said about the distinction between (H2) and (H3), or about the distinction between second and third level theories in general. The remarkable thing is that almost nothing has been said about it. Only recently have some psychologists discussed some closely related issues. For example, Zenon Pylyshyn says in a recent paper,

> By providing a principled boundary between the "software" and the "hardware", or between functions that must be explained mentally (i.e., computationally), and those that can only be explained biologically, one can factor the explanation into the fixed biological components and the more variable symbolic or rule-governed components. (1980, p.127)

But this passage is misleading in several respects which should be mentioned, if only briefly. First, it apparently presupposes that only "software" (i.e., "third level," program) operations are susceptible to computational explanation, whereas, in fact, computational accounts of direct or hardwired systems have been around longer than program using systems and their software have. If we want to explain the operation of a complex network of logic circuits, for example, we might well want to use a computational account. An electronic account would, in many cases, be exceedingly and unnecessarily complex. The second, related point to be made about Pylyshyn's remark is that the attribution of <u>any</u>

mechanism to explain a system's computation, whether "software" is used or not, involves some commitments about how that computation comes to be realized in the hardware, whether the hardware is biological, electronic, or whatever. The success of any computational account in psychology ultimately requires a biological explanation. And, in the third place, the presumption that all "hardwired" biological systems are "fixed" and unchanging certainly needs to be defended. At first blush, this view seems quite implausible -- surely our biological "hardware" changes quite a lot, and so it seems likely that any direct computational processing it is responsible for may also change and develop over time.

I have similar qualms about Pylyshyn's claim that "the architecture-algorithm distinction is central to the project of using computational models as part of an explanation of cognitive phenomena." (1980, p.126) Since algorithms can be computed directly, simply in virtue of the "architecture" of the system, the distinction he must have in mind is, I think, more appropriately named the "direct-program-using" distinction. And it should be noted that although there are clear cases, this distinction is not really quite distinct. It is blurred by "hybrid" systems, and, more seriously, by the difficulties of specifying in any precise and principled way what ought to count as a "control mechanism" of a progam using system. (See the last chapter for a more thorough discussion of these issues.) Nevertheless, the direct-program-using distinction <u>is</u> crucially important for third level theories, and Pylyshyn is certainly aware of this. He is one of the few psychologists who have given it any attention.

The reason that this distinction has been neglected is perhaps suggested by the following passage by Marvin Minsky,

> It is generally recognized that the greatest advances in modern computers came through the notion that programs could be kept in the same memory with "data", and that programs could operate on other programs, or on themselves, as though they were data. It is perhaps not so widely understood that one can obtain the same theoretical (though not practical) power in machines whose top level programs cannot be so modified...(1967, p.201n)

So our fundamental point about the equivalence of various sorts of systems might have been overlooked partly because this latter point is not widely recognized, and perhaps also because of an implicit faith in the assumption that the considerations that so overwhelmingly favor "stored program" systems in so many of our engineering projects will also apply to other quite different systems. That is, it is typically very much more practical to build a programmable computer to execute our programs than it is to build a hardwired computer to execute them. Programmable systems are much more flexible, more "plastic." So perhaps we should assume that the brain also has program using systems rather than hardwired systems. This point deserves careful consideration; we will return to it in §5.5., below. Let's begin by considering some other ideas.

5.2. Productivity

The theory that represented rules must be part of any adequate model of speech recognition has dominated the field for some time. In a well-known paper published in 1962, Halle and Stevens consider the possibility that phonetic representations of acoustic input could be provided by a procedure which compares segments of an input signal with the entries of a phonetic "dictionary." As they point out, this approach looks hopeless:

> The size of the dictionary in such an analyzer increases very rapidly with the number of admissable outputs, since a given phoneme sequence can give rise to a large number of distinct acoustic outputs. In a device whose capabilities would even remotely approach those of a normal human listener, the size of the dictionary would, therefore, be so large as to rule out this approach. (1962, p.156)

They suggest that the way to avoid this problem is with some form of an "analysis-by-synthesis" procedure which analyzes the input by comparing it to synthesized rather than stored representations. But then they immediately and without argument make the assumption that the rules governing such synthesis must be stored in memory the way the dictionary would need to be:

> The need for a large dictionary can be overcome if the principles of construction of the dictionary entries are known. It is then possible to store in the "permanent memory" of the analyzer only the rules for speech production... In this model the dictionary is replaced by generative rules which can synthesize signals...
>
> It seems to us that an automatic speech recognition scheme capable of processing any but the most trivial classes of utterances must incorporate all of the features discussed above -- the input must be matched against a comparison signal; a set of generative rules must be stored within the machine; preliminary analysis must be performed; and a strategy must be included to control the order in which the internal comparison signals are to be generated. (1962, p.157)

It is really surprising how blithely the assumption of represented, stored rules enters this discussion and establishes itself. Perhaps the authors see an objection to all but the stored-program models that they have just failed to mention, but it is more likely that the alternatives just did not occur to them. No objection to the view that the rules for synthesizing the linguistic structures are computed but not represented suggests itself .

5.3. Multiple access

Zenon Pylyshyn(1980) has suggested three criteria for deciding whether a computational process is "governed by representations and rules." He says that we should describe a process as governed by representations and rules if it is, among other things,

(a) "functionally transparent,"

(b) "arbitrary with respect to natural laws," or

(c) "informationally plastic." (p.120)

We will consider each of these criteria in turn.

Pylyshyn says that "the transparency condition reflects the multiple availability of rules governing relations among representational states:"

> Wherever quite different processes appear to use the same set of rules, we have _prima facie_ reason for believing that there is a single explicit representation of the rules, or at least a common subprocess, rather than independent identical multiple processes. The case can be made even stronger, however, if it is found that whenever the rules appear to change in the context of one process, the other processes also appear to change in a predictable way...These cases argue even more strongly that the system could not simply be behaving _as if_ it used rules, but must in fact have access to a

symbolic encoding of the rules. (p.121)

Ironically, this passage indicate precisely the point that defeats the view that "multiple access" provides some reason to suppose that the rules are represented or encoded and used. Pylyshyn is exactly right to point out that in such a case we have reason to suppose that there is either a single representation of the rules "or at least a common subprocess." The problem is that he does not explain why the former of these two possibilities should be preferred . Wherever the control mechanism of a program using system transfers control to a "single representation of the rules," a direct or hybrid system could execute the rules automatically using a "common subprocess." This point is similar to the point that both program using systems and direct systems can compute programs that have "loops;" there is no program that one of these kinds of systems can compute that the other cannot. When there is "multiple access" it can be explained by assuming either a single represented program or else by assuming a single subprocess. So why does such a case provide any reason at all for supposing the rules to be represented? "Functional transparency" provides no ground for a third level theory because of the fundamental equivalence of these various kinds of systems.

5.4. "Nonlawlike" processes

The second case suggested by Pylyshyn in which we would have reason to suppose that a processs might be rule governed is that in which the state transitions or input-output relations of the process do not "instantiate a natural law." He says,

> For example, there can be no nomological law relating what someone says to you and what you will do, since the latter depends on such things as what you believe, what inferences you draw, what your goals are...and, perhaps even more more important, how you perceive the event...Systematic but nonlawlike relations among functional states are generally attributed to the operation of <u>rules</u> rather than natural laws. (p.121)

It should be clear that the proper objection to this view is that it mistakenly supposes that the use of represented rules can bring about state transitions or input-output relations that other mechanisms cannot bring about. The fundamental equivalence of direct and program using systems has been neglected; any input-output relation that is the result of computation by a program using system could also be the result of computation by a direct or hybrid system. This much is clear regardless of what is made of the idea that some input-output relations are "nonlawlike" or "analog."

5.5. "Informational plasticity"

The third of Pylyshyn"s suggestions is that if a process is very "plastic," then it is probably rule-governed. He says,

> The informal notion of informational plasticity refers to the property of a system in virtue of which certain aspects of its behavior can be systematically altered by information, and hence by how the organism encodes stimuli or interprets events. We argued that the explanation of such behavior should appeal to rules and representations. (p.127)

Once again, it seems to me that what Pylyshyn is calling "plasticity," this malleability of the computational process, does not by itself provide any indication that there are represented rules governing the computation. Consider some hardwired system, like a simple electronic calculator. The operation of this system might seem very "plastic" in that its operation can be influenced in very short order simply by pressing various buttons. This is not because the calculator is a program using system, but simply because it is a system whose operation depends on what numbers, what data, is represented in its memory registers. One could imagine more complicated hardwired systems in which the relations between what is represented in memory and what is computed are more subtle. In these cases the plasticity does derive from the variability of the representations, but these are not representations that govern the computation in the way that a program governs a program using system. So plasticity does not generally support any third level claim to the effect that any process is governed by represented rules.

Suppose, then, that there were some creature who could learn a human language simply by looking at a transformational grammar of the language for a few minutes.[3] This "plasticity" in the creature's linguistic abilities would certainly cast doubt on the view that the creature was directly computing hardwired procedures with the grammar embedded in them in his use of the language. The reason is clear: it is not plausible that the direct system that computed the grammar could be built in such a short time. So we would reject this direct computation theory in

[3] This example was suggested to me by Noam Chomsky (personal communication).

favor of the view that the grammar is somehow represented in the creature and that this representation influences the computations carried out in the exercise of the creature's linguistic abilities. But this account is, so far at least, entirely neutral with regard to the question of whether the grammar is itself executed or whether it is just "data" that influences the computation of programs that are computed by the system. The plasticity of the process does not distinguish these two alternatives, though of course other considerations might do so.

Plasticity with respect to certain input, then, does support the view that the input is represented and somehow influencing the system, even if it does not itself support a third level theory about the processing involved. So plasticity is certainly of interest when we are developing a computational account of a system. Unfortuantely, it is not found in the sorts of human language processing that the grammar might be responsible for, at least as far as I know. The acquisition of linguistic competence in humans takes more than a few minutes. We can learn a word in a few minutes or less, but this is not the acquisiton of information that anyone assumes to be represented in the grammar of the language.

5.6. Neurophysiology

It is perhaps possible, at least in principle, that neurophysiological evidence could support a third level theory. No such evidence has been found, however. The neurophysiologist

David Hubel says in a recent paper,

> The brain does not depend on anything like a linear sequential program; this is at least so for all the parts about which something is known. It is more like the circuit of a radio or television set, or perhaps hundreds or thousands of such circuits in series or parallel, richly cross-linked. The brain seems to rely on a strategy of relatively hard-wired circuit complexity with elements working at low speeds... (1979, p.46)

But the distinction between hardwired and program using systems is hardly one that would stand out on casual inspection of the hardware. The distinction is really one of detail; one would need quite a thoroughgoing computational account of the neurophysiological mechanisms to settle this question. One would expect the computational account of the processes to be very well developed before the neurophysiology could be brought to bear on anything like this. In the area of language processing the underlying neurophysiological mechanisms are only very poorly understood, but even in very simple cases, it would be surprising if neurophysiological evidence would be of value in deciding between a second and third level account.

6. In sum, our discussion has revealed only one kind of presently accessible data that would clearly support the claim that the grammar is represented, viz., plasticity. But grammatical competence apparently does not exhibit the plasticity that would support such a claim. And even if it did, and if this led us to conclude that the grammar is represented, this evidence would still be neutral with regard to the question of whether the grammar was part of a program that is computed by a program using

system. Plasticity is what programmable systems are famous for, though. It is largely because of this that one hardly ever thinks of anything but a stored-program system when one considers executing a nontrivial program. Thus although plasticity by itself does not support a third level theory, still it is a feature that would be of interest to proponents of such a theory.

It is remarkable, then, that linguists typically do not mention plasticity in their discussions of (H1). It really begins to look like they must not have intended to propose (H3) at all: not only does the evidence offerred in support of (H1) fail to support (H3), but also, evidence that clearly would be relevant to the truth of (H3) is not considered. We can add to this the observation that many of the linguists' comments about (H1) indicate that they do not have (H3) in mind. For example, Chomsky compares (H1) to the hypothesis that a missile is controlled by a computer with a representation of a physical theory, without suggesting that the represented theory is somehow part of the program computed by the computer.(1980b, p.11) So the question is: If proponents of (H1) do not have (H3) in mind, then what <u>are</u> they proposing?

We have reviewed the evidence offerred by linguists in support of the hypothesis (H1) that the grammar G of a language is mentally represented and used in language understanding. It was argued that this evidence does not support the hypothesis,

(H3) In human language understanding, a program P which includes G as a proper part is represented and computed by a program using system.

The problem was that the evidence offerred in support of (H1) does not serve to distinguish (H3) from,

> (H4) In human language understanding, a program P which includes G as a proper part is computed but not represented; it is computed by a hardwired or hybrid system.

In the light of the linguistic evidence, (H4) seems at least as plausible as (H3), if not more so. We discovered no evidence offerred for (H1) which supported (H3) any more than it supported (H4).

So suppose that proponents of (H1) do not mean to porpose (H3). What else might they mean? Well, they might be construed as intending,

> (H5) In human language understanding, a program P is computed (by either a program using system or by some other kind of system) which uses G <u>as</u> <u>data</u>.

This hypothesis claims that under some interpretation, a language processing system has a representation of G in memory to which certain computational processes are sensitive; parts of G are taken as argument to functions that are computed. But proponents of (H1) are never so explicit about the role of the grammar, so it is really more natural to assume that they are proposing that either (H3) or (H5) is correct. That is, the most natural idea is that they mean to suggest that whatever model of speech processing is correct, it will be one in which the grammar is represented and used <u>somehow</u>. We can formulate this view as follows:

> (H6) In human language understanding, the program computed either includes G and is computed by a program using system <u>or</u> uses G as data <u>or</u> both.

This proposal seems rather unnatural, but, as was observed above, a proposal of just this sort is what evidence of plasticity would support.

So now the question is whether the evidence adduced in support of (H1) supports (H6). Does the evidence for (H1) serve to support (H6) rather than (H4), for example? We did not consider this looser hypothesis, (H6), but a quick review of our discussion suffices, I think, to show that (H6) will fare no better than (H3) did. The strategy against (H3) was to argue that in the light of the evidence, (H4) was just as plausible as (H3), so there is no reason to prefer the third level theory. If we succeeded in making this case for the plausibility of (H4), then (H6) is undermined along with (H3); the evidence supports the non-representational models just as well as it supports the representational models. So it is hard to make sense of the linguists' methodology whether they are proposing (H3) <u>or</u> (H6). Furthermore, since (H6) leaves open the possibility that no program using system is involved in language processing, it is hard to see what can be made of Chomsky's claim that language behavior is rule-governed. Surely we do not want to say that <u>any</u> second level system whose memory set includes rules exhibits rule-governed behavior.

So what do proponents of (H1) mean to claim? Given any program, there are indefinitely many different kinds of computing systems that can compute that program. Some of these systems are governed by a representation of the program; some of them are not; and in others it may be unclear which account is correct. In the case of language understanding, it is still a matter for speculation what sort of program is computed, if a computational account is appropriate here at all. The issues involved in deciding among the various computational accounts that do or do not suppose that the grammar is represented are really quite complex. It seems to me that the psychology of language is not yet ready to confront them in any serious way. I do not think that the linguists come to grips with them either, yet the idea that grammars (or language processing algorithms) are "mentally represented" has become firmly entrenched.

Even if there is no correct model of human language processing that attributes mental representations of grammars to speakers of human langauges, this would not render linguistic theory devoid of content! That is an absurd idea. The hypothesis (H4) has it that the grammar is not represented but nevertheless gives the grammar a central role in the theory's decription of language processing. Or it could be that the lannguage processing algorithms (which may or may not be represented themselves) capture the information represented by the grammar in some different form. There may even be algorithms that derive the language processing algorithms from the grammar, as some psychologists and linguists have suggested. The possibilities are still wide open.

In the passage quoted at the head of this paper, Richard Feynman notes how some scientists objected to Newton's theory because it did not give an explanation of the mechanism of gravitation. Now it is clear how ridiculous it was to think that this objection should call the significance of his achievement into question. I think that the situation in linguistics is analogous. The fact is that linguistic theory gives an interesting and largely correct account of an impressive body of data. If we do not yet have any idea of what the mechanisms of language processing are, that hardly detracts from the linguists' achievement.

BIBLIOGRAPHY FOR CHAPTER IV

Braine, M. (1971) On two types of models of the internalization of grammar. In D.I. Slobin, ed., *The Ontogenesis of Grammar*. New York: Academic Press.

Chomsky, N. (1969) Comments on Harman's reply. In S. Hook, ed., *Language and Philosophy*. New York: New York University Press.

Chomsky, N. (1972) *Language and Mind*. New York: Harcourt Brace Javonovitch.

Chomsky, N. (1973) Conditions on transformations. In S.R. Anderson and P. Kiparsky, eds., *A Festschrift for Morris Halle*. New York : Holt, Rinehart and Winston.

Chomsky, N. (1975) *Reflections on Language*. New York: Pantheon Books.

Chomsky, N. (1977) On Wh-movement. In P.W. Culicover, T. Wasow, and A. Akmajian, eds., *Formal Syntax*. New York: Academic Press.

Chomsky, N. (1980a) *Rules and Representations*. New York: Columbia University Press.

Chomsky, N. (1980b) Rules and representations. *The Behavioral and Brain Sciences*, *3*, pp.1-61.

Chomsky, N. (1980c) On binding. *Linguistic Inquiry*, *11*.

Culicover, P.W., Wasow, T. and Akmajian, A. (1977) Introduction. In P.W. Culicover, T. Wasow and A. Akmajian, eds., *Formal Syntax*. New York: Academic Press.

Feynman, R. (1965) *The Character of Physical Law*. Cambridge, Massachusetts: MIT Press.

Fodor, J.A., Bever, T.G. and Garrett, M.F. (1974) *The Psychology of Language*. New York: McGraw-Hill.

Halle, M. and Stevens, K. (1962) Speech recognition: a model and a program for research. IRE Transactions on Information Theory, IT-8, pp.155-159.

Hubel, D.H. (1979) The brain. Scientific American, 241(3), pp.44-53.

Johnson-Laird, P.N. (1977) Procedural semantics. Cognition, 5, pp.189-214.

Kaplan, R. and Bresnan, J. (in press) A formal system for grammatical representation. In J. Bresnan, ed., The Mental Representation of Grammatical Relations. Cambridge, Massachusetts: MIT Press.

Lasnik, H. (in press) Restricting the theory of transformations: a case study.

Marr, D. and Poggio, T. (1976) From understanding computation to understanding neural circuitry. MIT Artificial Intelligence Memo 357.

Miller, G.A. and Chomsky, N. (1963) Finitary models of language users. In R.D. Luce, R. Bush and E. Galanter, eds., Handbook of Mathematical Psychology, Volume II. New York: Wiley.

Minsky, M.L. (1967) Computation: Finite and Infinite Machines. Englewood Cliffs, New Jersey: Prentice-Hall.

Pinker, S. (in press) A theory of the acquisition of lexical-interpretive grammars. In J. Bresnan, ed., The Mental Representation of Grammatical Relations. Cambridge, Massachusetts: MIT Press.

Pylyshyn, Z.W. (1980) Computation and cognition: issues in the foundation of cognitive science. The Behavioral and Brain Sciences, 3, pp.11-169.

Searle, J.R. (1980) Rules and causation. (Commentary on Chomsky, 1980b.) The Behavioral and Brain Sciences, 3, pp.37-38.

Stelmach, G.E., ed. (1978) *Information Processing in Motor Control and Learning*. New York: Academic Press.

Von Neumann, J. (1958) *The Computer and the Brain*. New Haven: Yale University Press.

Young, J.Z. (1964) *A Model of the Brain*. Oxford: Clarendon Press.

CHAPTER V

LANGUAGE LEARNING AND INNATE CONCEPTS

1. Theories in computational psychology typically assume that an organism, or a physical system within an organism, can be said to "have a certain concept" if it uses some internal symbol which, under an appropriate theoretical interpretation, expresses that concept. All the theories that will be considered in this chapter rest on some such assumption. Given such an assumption, it is safe to say that computational theories of natural language processing typically assume that in order to learn or understand a predicate, a language processing system must either have the concept expressed by that predicate or else have concepts in terms of which that predicate can be defined. Hence, the concept expressed by any learnable predicate must be either an unlearned, "primitive" concept or definable in terms of the unlearned primitives, and so theories of concept learning typically attempt to explain how new concepts are constructed out of the primitive basis by definition. The question of how large this primitive basis must be, of what it must contain to account for human language processing, is a point of current controversy.

In a number of recent papers, Jerry Fodor and his associates have argued that since there is good reason to believe that most lexical items do not have non-trivial definitions, attempts to explain how lexical concepts (i.e., concepts expressed by lexical items) are constructed from more basic concepts must fail, and so most lexical concepts must be unlearned, innate primitives.[1] In this chapter it will be argued that both sides of this controversy have been led astray by an untenable assumption, viz., that concepts must either be definable or else unlearned and innately given. An alternative theory will be proposed that can adopt the best features of the rivals in this controversy because it rejects this assumption; it is a clear way through the middle of the dispute. I will argue that it is, in fact, the only plausible theory of language and concept learning we have.

2. The traditional approaches to language processing

2.1. Introduction

How does one learn a term or predicate of a natural language? All of the plausible theories agree that learning a word involves framing and testing hypotheses about the semantic properties of the word. That is, on the basis of some evidence, the language learner draws some conclusions about the word, conclusions which are confirmed but not entailed by the evidence and which are typically subject to revision in the light of later evidence. Some theories can be construed as maintaining that to learn some predicate P of a natural language, one must actually formulate and confirm a true hypothesis of the form,

(H) For all x, x is in the extension of P if and only

[1] The relevant papers include Fodor(1975), Fodor et al.(1975), Fodor(1980), Fodor et al.(1980) and Fodor(ms).

if Gx,

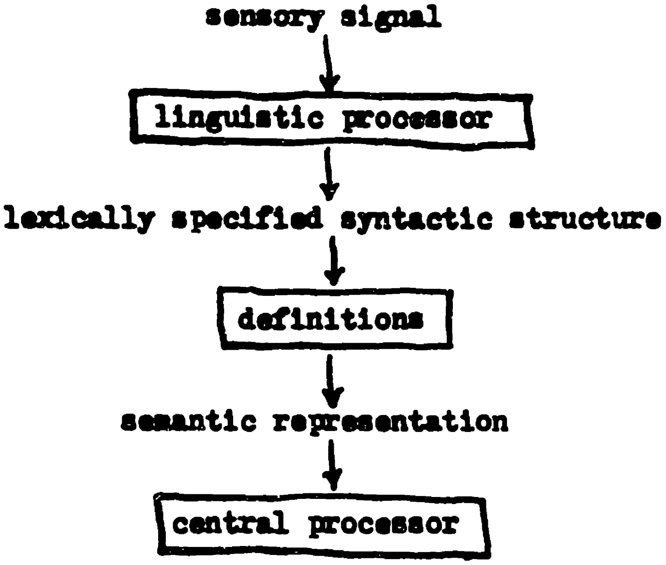
or some equivalent hypothesis.[2] The theories that propose some such account divide on the character of the concepts expressed by the term in the place of 'G' in such hypotheses. We will call a learning theory Empiricist if it maintains that the lexical items of natural languages are defined in terms of some set of "primitive" concepts notably smaller than the set of lexical concepts, so that learning a word involves learning a definition of the word in terms of these primitive concepts. Thus the Empiricist holds that the term in the place of 'G' is typically a complex expression whose constituents express primitive concepts. The opposing view is that of the Nativist theories which assume that the set of concepts needed to frame such hypotheses is virtually the same size as the set of lexical items, so that a word is typically learned by associating it with the appropriate primitive concept. On the Nativist account, then, the term in the place of 'G' is typically a simple expression which expresses a non-definable lexical concept.

This division among theories of language learning is related to a similar distinction among theories of language understanding. We will say that a theory of language understanding is a Deep or Decompositional theory if it holds that understanding a linguistic expression involves replacing the

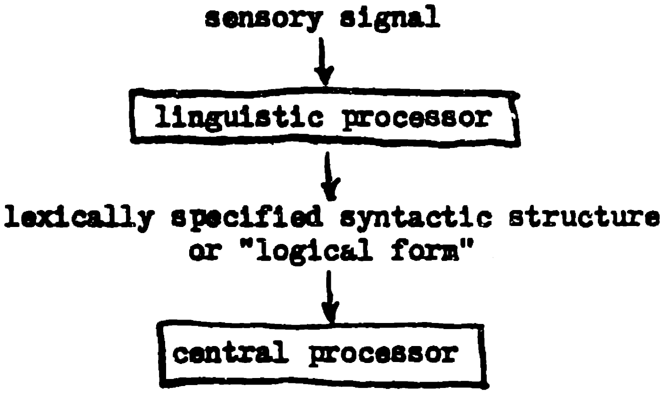
[2] Notice that the predicate in the place of 'G' is used, while in the place of 'P' we will have the name of the predicate we want to mention.

lexical items of the expression with their definitions to formulate a "semantic representation" of the linguistic input. Shallow theories, on the other hand, do not assume that definitions are retrieved in understanding. The difference between these two kinds of theories can be seen in the following diagrams:

(FIGURE 1) A Deep Theory



(FIGURE 2) A Shallow Theory



Theories of either kind may want to allow "top-down" influences of the "central processes" on the operation of the "linguistic processor," and this may blur the distinction between the two processors represented in each diagram. And, of course, the central processor must have access to more information than is provided by a "semantic representation" or "lexically specified syntactic structure." In the case of verbal input, for example, the central processes must have some access to information about stress, tone of voice, etc. But such elaborations do not obscure the basic difference between Deep and Shallow theories: Deep theories are distinguished by their supposition that at some stage in the processing of linguistic input most lexical items are replaced by their definitions.

A Nativist holds that lexical items typically do not have non-trivial definitions, so he will have have to elect a Shallow theory of language understanding.[3] If there are no definitions, it can't be that definitions are recovered in understanding linguistic input. The Nativist must assume that the central

[3] I have tried to divide up the positions along standard lines, but one must be careful about the terminology. Jerrold Katz, for example, is not a Nativist who has been inclined towards a Deep theory. Despite the fact that he accepts strong nativist commitments to innately given phonological, syntactic and semantic universals, he is an "Empiricist" in the sense specified above, because he apparently would hold that in learning a word we typically learn that it expresses a complex concept built up from primitive semantic features. So, in our terms, he is (or at least he was in Katz, 1966, 1972) an Empiricist who elects a Deep theory of understanding. He rightly wants to distinguish himself from empiricists who do not accept his nativist commitments and especially from those who think that claims about meaning must be reducible to claims about observable behavior. Our terminology, however, does not mark these distinctions.

processes are defined over the surface morphological inventory of the language. An Empiricist, on the other hand, can elect either a Deep or a Shallow theory. He may want to maintain that the definitions of lexical items must be recovered in understanding, or he may want to maintain that definitions are only accessed by the central processes in response to certain task demands.

2.2. Problems for Empiricist and Deep theories

The most serious problem for Empiricist and Deep theories is their inability to provide non-trivial definitions for lexical items, i.e., definitions that are, unlike most dictionary definitions, strictly and necessarily true. Attempts to define lexical items in terms of some notably reduced vocabulary have been notoriously unsuccessful, and there have been a number of various and sustained attempts of this sort in the philosophical tradition. No one has ever been able to discover the sense data language that traditional empiricists would have liked to ground their epistemology, and the failure of the positivist program in the philosophy of science is largely due to a similar inability to reduce theoretical claims to observation claims. But the Empiricist account of language learning really begins to look implausible when one considers the scarcity of non-trivial definitions even when the defining vocabulary is <u>not</u> restricted. And if any further objection were needed, there are the well-known attacks on the analytic-synthetic distinction which strike at the very foundation of the Empiricist program. If there are no analytic definitions, then it cannot be that we learn a word by defining it in terms of concepts drawn from some restricted set. And, of course, if there are no definitions, it cannot be the case that we must recover definitions in understanding our language.

There is also psychological evidence against the Deep theories of language understanding. For even the best cases of definable lexical items, like 'bachelor' or 'kill', attempts to find parameters of the language users' responses that are sensitive to properties of the definitions of the expressions being defined have been uniformly unsuccessful.[4] Furthermore, the evidence shows what everyone would suspect already, that understanding takes place very quickly, so, ceteris paribus, a theory that requires less processing between input and output should be preferred.[5] If the recovery of definitions is not required, if it does not increase the efficiency of the processing, then we should not asssume it is involved. The psychology of language can offer nothing like a complete account of these data, but what evidence there is certainly seems to favor a Shallow account of language understanding.

2.3. Problems for the Nativist

Unfortunately, there are reasons for thinking that the Nativists' account of language learning is not right either.

[4] See Kintsch(1974), Fodor et al.(1975), and Fodor et al.(1980).

[5] See, e.g., Marslen-Wilson(1973).

Their best known problem is their rather extreme view about what must be innate, a view which follows closely on their denial that lexical items typically have non-trivial definitions. If a term or predicate has no definiton, then one must already have the concept expressed by the expression before the expression can be learned. For example, if learning the term 'electron' involves confirming a true hypothesis like the following,

> (H1) For all x, x is in the extension of 'electron' if and only if Fx,

then one must already have the concept _F_, i.e. the concept _electron_, before the term can be learned. This view might seem compatible with the commonsense view that learning a word typically goes hand-in-hand with learning the corresponding concept, that one's use of a word depends on the concept it is being taken to express, but the Nativist cannot accept this commonsense view. His claim that terms typically do not have definitions forces him to reject the view that lexical concepts are typically learned.

How could someone learn the concept _electron_? The only plausible theories of concept learning are hypothesis testing theories, but if concept learning, like word learning, involves confirming a true hypothesis about the extension of the concept, then one cannot learn a lexical concept unless one already has it. But this is absurd; one cannot learn what one already knows. The Nativist is thus driven to the conclusion that virtually all lexical concepts are unlearned. The _learning_ of a concept, the confirmation of a true hypothesis about the extension of the concept, would presuppose the prior availability of the concept to be learned. Since we do have concepts, then, they must be innate or triggerred, _not_ learned.

Fodor calls this result "a pretty horrendous consequence." (1975, p.80) It is, at least, a shock. The result certainly conflicts with ordinary preconceptions about the matter, but preconceptions are not always correct. Is there any more substantial reason to assume that this shocking result is really incorrect? It is important to keep in mind here that the Nativist does not need to assume that children spring from the womb with fully developed conceptual systems, already masters of the concepts of particle physics and music theory, just waiting to get them "hooked up" to the language and the world in the right way. The Nativist _can_ allow that the child's conceptual system is altered by experience. He is only committed to the view that whatever concept acquisition there is is not concept _learning_. It is not the confirmation of a hypothesis about the semantic properties of the concept. As Fodor says,

> An environmentally occasioned alteration in the...conceptual system counts as a concept learning experience only if what is learned (under its theoretically relevant description) stands in a confirmation relation to the events which cause it to be learned (under their theoretically relevant descriptions). That is, it's concept learning only if it involves the projecting and testing of hypotheses. (1975, p.87)

The Nativist needs to reject the view that concepts are acquired by learning in _this_ sense. Is there any real reason, then, to suppose that any undefinable concepts are learned, not triggerred or innately given?

It seems that there is. Presumably the Nativist does not want to maintain that I had the concept electron before I ever heard a word of chemistry or physics. But if the Nativist accepts the view that somehow my study of physics and chemistry was the occasion of my acquiring the concept, he is left in the unappealing position of having to maintain that there is no theoretically relevant description of these experiences under which they serve to confirm what I learned when I learned the concept. Surely the evidence is such as to suggest that there is likely to be a confirmation relation here; the content of my experience in physics and chemistry was crucially relevant to my acquisition of the concept. For this reason the acquisition of the concept looks like learning, and for this reason it looks like just the sort of thing that an information processing theo should be able to capture. The challenge for this intuitive view is to explain to the Nativist how concept acquisition could be learning if the concepts typically do not have non-trivial definitions. A consideration of some problems that are common to both Nativist and Empiricist theories will prepare the way for taking up this challenge.

3. The interpretation of internal representations

The Empiricists and the Nativists agree that learning a predicate or a term P involves confirming a true hypothesis equivalent to one of the form,

(H) For all x, x is in the extension of P if and only if Gx.

If this theory is literally true, such a hypothesis must be internally represented and subject to evidence bearing on its truth. The confirmation of such a hypothesis will in effect associate the internal representation of the term P with some internally represented term 'G' which expresses the concept expressed by P. Since a representation has content or expresses something only under some interpretation, the intended claim must be that the internal term 'G' expresses the appropriate concept under the theoretically relevant interpretation. This raises the question, then, of how the theoretically relevant interpretation of the internally represented term is determined. What is it in virtue of which the internally represented term expresses a particular lexical concept?

The only reasonable answer to these questions that has been proposed is that the appropriate theoretical interpretation of an internal representation is determined, or at least partially determined, by its causal role, i.e., by its relations to inputs, outputs and other states.[6] For example, an internal state that is appropriately interpreted as a representation expressing the concept of the color _red_ would presumably be one that plays a role in the representation of certain information about what is visually perceived, one that is associated with internal representations of color words of the subject's spoken language, one that plays a role in the production of verbal and graphic

[6] See, e.g., Chapter I; Loar, ms; Fodor, ms-a.

reports about red things, and so on. There is no question but that this is terribly vague, but it would not be expected that a detailed account would be provided before the theory of internal processes was developed. On the contrary, it is just such a theory that will tell us what role an internal representation must have in order that it allow a certain theoretically relevant interpretation. Or at least this is what psychologists are apparently presuming. No other credible view has been proposed.

There is nothing like a complete theory of internal processes available, but there are some vague suggestions about how an internal representation of a concept will be related to representations of other concepts. Some Empiricists and Deep theorists assume that the internal representation of the concept bachelor will be related by definition to the internal representations of the concepts unmarried and male, for example. And some Nativists suggest that although there may not be any definitions, there are "meaning postulates" that "mediate whatever entailment relations between sentences turn upon their lexical content."(Fodor et al.,1975) Meaning postulates do not serve to eliminate lexical items in favor of their definitions, but specify semantic relations between lexical items (or between internal terms expressing the corresponding lexical concepts) for use by "central processes." Some may have the biconditional form of definitions, but others will be simple conditionals, as in,

For all x, x is red only if x is colored,

or,

For all x, y, x causes y to die only if y dies.

Certainly it seems plausible that the information represented by such rules is used in natural language processing. In any case, the processing system is presumed to respect and make use of such represented information. Thus the internal representation of a term has the causal role it does partly in virtue of these represented relations to other terms.[7] The theorist will want to choose an interpretation of the internal states of the system that is "appropriate" given such considerations.

These proposals obviously do not go very far towards a specification of the required interpretation of the internal states and events. When psychologists propose that learning a term involves confirming a hypothesis equivalent to an instance of (H), they are apparently just presuming that a specification will be forthcoming of what role a state must have to be "appropriately" or"correctly" interpreted as a representation expressing the relevant concept. One might be sceptical about this presumption, but the strategy of proposing such computational accounts of how representations must be interrelated and used in processing has proven to be fruitful. What further encouragement could be expected at this point?

----

[7] This point is emphasized in the Artificial Intelligence literature as is not surprising. See, e.g., Miller and Johnson-Laird, esp. pp.127-128; Woods, 1975, esp. pp.42-43. This point is the computational version of the philosophical view that the meaning of a word is determined by its role in perception, inference and action. See, e.g., Harman, 1975, esp. pp.294,283-284.

There is another issue concerning the interpretation of internal representations which should be considered. Some philosophers have argued that learning the meaning of a word is, at least in many cases, not just a matter of associating it with an appropriate internal representation. They argue that the question of whether one knows the meaning of a word, or has the concept expressed by that word, depends in part on one's physical and social environment. For example, Hilary Putnam(1975, ch.12) argues, in effect, that someone can have the concept <u>water</u> only if he or someone in his society has had some interaction with water, with H_2O. A typical English speaker knows the word 'water', has the concept and has beliefs about water, but a physically identical individual who had never seen or had any interaction with water but instead had had interaction with some other chemical, "XYZ," cannot be correctly said to have the concept <u>water</u> or to know the meaning of the word 'water'. This other individual would presumably have beliefs about some liquid he calls "water," i.e., XYZ, but these are not beliefs about "water" in our sense of the term, or so it is claimed. Tyler Burge(1979) urges that the subject's physical and social environment is crucially relevant to the question of whether a person has a particular concept in those cases where the subject has the concept but does not have an expert's grasp of it. (See also Evans, 1973.) It is beyond the scope of this paper to consider the arguments for these claims in detail, but since they are persuasive and widely accepted, it is important to consider how they bear on Empiricist and Nativist theories of language learning.

Suppose that Putnam is right about the concept _water_. The Empiricists and Nativists could still hold that learning the word 'water' involves associating it with an appropriate internally represented term, but they would presumably want to give up the view that the only appropriate interpretation of this internal term is one under which it expresses the concept _water_; in some other social and physical context the same term would express the concept _XYZ_. Learning the meaning of the word 'water', then, could not be simply associating the word with the appropriate internal representation. This sort of point might well make a psychologist worry about how far such indeterminacy can be pushed, but I think that it is clear that the psychological theories are not seriously threatened. That is, the theories are still significant and of interest even if there is considerable indeterminacy in the proper interpretation of internal representations.

It is of interest to account for the causal role of the internally represented term associated with the word 'water' by a typical English speaker, for example, even if the appropriate interpretation of this internal term in some other circumstances might be something quite different. Even if a theory is "bound" in this way to our physical and social context, its results will still be interesting. The computational theories aim to provide a formal account of the internal computation, an acccount of the relations between internal states, inputs and outputs, and this part of the theory is context-free; only the interpretation is context-dependent. But the formal account will be significant in itself. If psychologists could produce only this sort of account, only a formal account of the language processing and other internal processes, their achievement would not be wanting for importance!

In sum, it seems that there is no objection to developing context-dependent theories in psychology. Of course, it will be desirable to generalize these theories as it becomes clear how to do so, but in the meantime there are apparently no grounds for concern over the integrity of computational theories.[8] However, as we will see, one of the motivations for Empiricist and Nativist theories appears to rest on a neglect of the role of context in the determination of the appropriate interpretation of internal representations.

4. The Cognitivist theory of language learning

Now the Nativist's challenge will be taken up; a theory of language learning will be suggested that gives an account of how words and concepts can be learned even when non-trivial definitions are not available. This alternative theory will be called the "Cognitivist" theory.

[8] See Chapter II for a more general discussion and defense of this point.

The best versions of the Empiricist and Nativist theories have a good deal in common. The Nativist theories are roughly what you get when the commitment to defintions in the Empiricist theories is rejected. They share the following basic assumptions:

(1) The organism has a set of primitive, unlearned concepts.

(2) Learning a word involves framing and testing internally represented hypotheses.

(3) These hypotheses are, in effect, tenative definitions of the word in terms already available to the internal processes.

(4) Therefore, all concepts that the organism can learn are definable in terms of the primitive, unlearned concepts.

(5) Therefore, the set of concepts potentially available to the organism is determined by its innate endowment.

If neither Empiricist nor Nativist theories look plausible, it could well be that there is an error somewhere among their shared asssumptions. The Cognitivist urges that this is, in fact, the case. The source of the problems is (3). The Cognitivist agrees with the other approaches that (1) and (2) must be accepted, but he urges that the most plausible approach rejects (3) and its consequeces, (4) and (5). Let's consider how the Cognitivist proposes to avoid assumption (3).

The Cognitivist gives an account of language and concept learning that parallels the ordinary, intuitive account much more closely than the Nativist and Empiricist theories do. Suppose that the term 'elm' is undefinable; that there is no strictly and necessarily true non-trivial definition of this term. This is plausible. And suppose that we have a subject who does not have the concept elm and has never seen or heard the word before. How could such a subject learn the word? Well, in an ordinary situation he might learn the word by hearing someone else use the word and asking what the word means. He might get an answer like, "An elm is a deciduous tree with simple leaves that have jagged edges." If this is not enough to convey the concept, suppose that the answer is as thorough as you like. What internal information processing occurs in such a situation? At first, the subject identifies the word. Let's suppose that he associates some internally represented term, 'ELM', with the word's phonological, morphological and syntactic descriptions. Then he begins to accumulate information about the extension of the word. Let's use the term 'semantic information' very loosely to apply to any such information; it may be partly non-linguistic, empirical and even inaccurate. If he is told that an elm is a deciduous tree, and he sees no good reason to doubt this information, we can assume that he internally represents the fact that 'elm' and hence also 'ELM' have in their extensions only things that are deciduous trees. This process is learning, since the subject's experience serves to confirm the truth of the information represented, and this represented information may be open to later disconfirmation.

How does the subject learn the concept expressed by the word 'elm'? Again, learning a concept is assumed to involve nothing more than learning some ordinary things about the extension of the concept. It is in virtue of having represented such learned information that the relevant internal term comes to have the appropriate causal role, i.e., the causal role in virtue of which it is properly interpreted as expressing the the concept. Thus the Cognitivist is not committed to the absurdity that a concept must already available before it can be learned. Rather, the concept becomes available through the learning. The subject may well have a suitable collection of learned information about the extension of the concept without ever having heard the word, in which case, of course, the subject would know the concept but not the word.

The basic idea of the Cognitivist approach is this commonsense idea that at some point in learning some ordinary information about a word the subject can correctly be said to know the word. He will have learned enough that the associated internal representation expresses the concept expressed by the word. <u>This</u> is how a word typically comes to be associated with the appropriate internal term, and hence with the appropriate concept. It is <u>not</u> generally the case that this association is established by confirming an internally represented hypothesis equivalent to an instance of (H). Rather, in learning certain facts about the word, internally represented information typically leads to certain conclusions about the word, conclusions other than instances of (H). Similar views can be found in the philosophical literature. Hilary Putnam says, for example,

> What I contend is that speakers are required to know something about (stereotypical) tigers in order to count as having the word 'tiger'; something about elm trees (or anyway, about the stereotype thereof) to count as having acquired the word; etc.
>
> ...the "information" contained in stereotypes is not necessarily correct...Most stereotypes do in fact capture features possessed by paradigmatic members of the clas in question. (1975, pp.248,250)

The notion of stereotypes has also been used in the explanation of certain psychological data, as for example in the work of Eleanor Rosch. The Cognitivist account is not committed to the view that every concept has a stereotype, a particular body of information that must be mastered before one knows the concept, but this sort of suggestion is clearly in line with the Cognitivist program. The revolutionary idea is that learning a concept comes about by learning things that are nothing at all like definitions of the concept. This has been the commonsense doctrine all along, but it is rejected by the Nativists and Empiricists. Let's consider why they think that this view cannot be right.

5.1. An ordinary language objection

The most obvious objection to the Cognitivist account of the learning of, say, the concept elm, is that it does not tell us how the concept is learned at all. It misses the mark entirely.

Instead of telling us how the concept elm is learned, it just gives the standard account of how some ordinary information about elms is learned. But that information is not the concept, sc where is the Cognitivist account of how the concept is learned? What account can be given of how the concept is learned that does not require the absurd assumption that the subject already has it? This is the hard question.

This objection is, I think, unsound, but it can be conceded without losing anything too important from an empirical standpoint. The Cognitivist proposes that there is nothing more to learning the concept than learning certain information which then has the appropriate role in internal processing and in the causation of behavior. It is true that this information is not itself the concept, but this is no objection even to the view that learning the information is necessary and sufficient for learning the concept. If one learns the rules of chess and acts according to them then one has learned the game; the fact that the rules are not themselves the game does not show that learning the game requires learning something more than just the rules.

So even if one takes seriously the objection that the information learned is not the concept and so learning the concept cannot consist in learning this information, this does not touch the Cognitivists claim to have presented an account of all the internal processes relevant to concept learning. That is, the Cognitivist can concede to the objector's point and still claim that once one has learned the relevant information, one has the concept. But of course Nativists and Empiricists are concerned not merely to make the conceptual point but also to reject this theory, so they have missed their mark.

5.2. Is Cognitivism really Empiricism?

There is another similar but more sophisticated objection to the effect that Cognitivism must really be either Empiricist of Nativist. THe best way to see how the Cognitivist avoids this dilemma is perhaps to keep in mind how a simple computational account of language learning might go. There are computer programs already in use that provide some crude indication of what might be involved. It is easy to write a program which can, on the basis of some user's input, associate a word like 'elm' or 'tiger' with a list of properties possessed by typical members of its extension.[9] The Cognitivist claims that something like this is involved in human language learning. The Nativist, however, is likely to respond that any such proposal is really just a particularly implausible incarnation of Empiricism. (See, e.g., Fodor, 1978.)

Suppose, for example, that a subject associates the word 'elm' with certain properties, such as the property that it has in its extension only things that are deciduous trees typically having leaves of a certain shape. The Cognitivist agrees with

[9] The programming language LISP has specific features to make it suitable for writing such programs. See Winston and Horn, 1981, Ch.5 or Greenberg, 1978.

the Nativist that this information does not provide anything like an analytic definition of 'elm', but nevertheless maintains that it typically suffices to give the subject the concept. What is the Nativist objection? Well, the Nativist would presumably object that the Cognitivist has no grounds for assuming that learning this information constitutes learning the concept *elm* rather than, say, the concept *mutant oak*, where a mutant oak is an oak that has elm-shaped leaves because of some minor genetic peculiarity. The Nativist, on the other hand, does have some basis for for distinguishing the learning of the one concept from the learning of the other; he just *assumes* that there must be something "built into" the language processing system that distinguishes the concept.

This Nativist assumption is, I think, implausible. In the first place, it is not at all clear that a system that operated the way the Nativist proposes could *do* anything that a Cognitivist system could not do. The Nativist does not want to assume, for example, that a subject has the concept *elm* only if he can correctly distinguish elms from every other kind of tree. And in the second place, the Nativist is unable to explain what it is that makes the concept *elm* different from the concept *mutant oak* or any other similar concept.

If the Nativist account of what distinguishes the concept *elm* from the concept *mutant oak* is incorrect, then what does distinguish these concepts? What *is* it in virtue of which the subject has learned the former but not the latter? And what distinguishes this subject from one who seems to have learned the word 'elm' but has actually associated the word with the concept _mutant oak_? Well, this case is very much like the ones considered by Putnam, Burge and others in their discussions of the relevance of social and physical context to ascriptions of mental states. When a subject learns the concept _elm_ it may well be that the only thing in virtue of which he has learned this concept rather than the concept of _mutant oak_ is his social and physical context. In the English speaking community he can use the word 'elm' to inquire about elms rather than about mutant oaks because there are experts around to decide any questionable cases. Or he may treat the concept as one that applies to those trees in his yard and others of the same kind, when those trees in his yard may, in fact, be elms. If the context is relevant in any such cases, as seems plausible, then the Nativist view that there must be something internal, something built into the language processing system to distinguish each learnable concept, can be avoided. The Cognitivist view seems much more plausible.

5.4. Cognitivism and a psychology of belief and desire

The Nativists and Empiricists will complain that the Cognitivist account is too vague, so vague that it becomes unclear whether there will be any fact of the matter whether a subject has any particular concept. Perhaps this is the real issue between Empiricists and Nativists, on the one hand, and Cognitivists, on the other. The motivation for the traditional assumption (3), the reason that the Nativists and Empiricists feel compelled to assume that learning a word involves confirming the hypothesis that the word expresses the concept that it does in fact express, is that they think this provides a correct, clear account of what must be learned to learn a word. On the Cognitivist view, on the other hand, it is very unclear whether any subject can be said to have the required concept. And this makes it look unclear that the Cognitivist is going to be able to reconstruct anything like a belief-desire psychology, since one cannot have a belief about something unless one has the concept of that thing (or so it is assumed).

This objection to the Cognitivist fails, however. In the first place, as has been urged above, it is not at all obvious that the traditional theories _are_ right about what must be learned when one learns a word. The Cognitivist account of what must be learned is much more intuitive and does not face the problems faced by the other approaches. In the second place, although it is true that the Cognitivist has not provided any clear account of when an internally represented term can correctly be said to express a particular concept, he is no worse off in this respect than the Nativists and Empiricists. As was noted above, neither the Nativists nor the Empiricists nor anyone else has a clear account of what it is in virtue of which an internally represented term gets the appropriate interpretation. They have offerred vague and partial accounts. For example, as was noted above, in cases like that of the word 'bachelor' it is plausible that one does not know the word unless one knows the definition. If this _is_ a requirement then the Cognitivist can also say that one has not learned the word until this definition has been learned. Similarly for the Nativists' meaning postulates; if they are what individuates a concept, they are what must be learned before one can be said to have the concept. On the Cognitivist account, the knowledge required in such cases is typically learned, not innate. So wherever the Empiricists and Nativists can be precise, the Cognitivist can be precise too. The Cognitivist approach does not open up any <u>new</u> dangers for the reconstruction of a viable concept of <u>concept</u> that could serve in a belief-desire psychology. And finally, we have urged above that the Nativist and Empiricist theories are in fact worse off in regard to this project of specifying the appropriate theoretical interpretation of internal representations than the Cognitivist is. Their assumption that there must be something built into the language processing systems in virtue of which their terms express the concepts they do is implausible; contextual factors are surely relevant in many cases.

6. Advantages of Cognitivism

The Cognitivist theory is just as compatible with the Shallow theory of language understanding as the Empiricist and Nativist theories are. It assumes (roughly) that a spoken word is mapped into some symbol of the language understanding system which expresses the concept expressed by the word. The Cognitivist rejects assumption (3), however; i.e., he rejects the view that learning a word always involves confirming a rule that associates the spoken word with the internal term. In rejecting this assumption, the Cognitivist approach immediately acquires an impressive list of advantages over its rivals. It is supported by the commonsense, intuitive account of what goes on in language learning. It is not committed to the view that most lexical items have definitions as the Empiricist is. It can provide a natural account of concept learning, so it does not need to hold that virtually all lexical concepts are innate or triggerred as the Nativist does. And finally, since learning a term does not typically involve defining it in terms already available to the system, assumptions (4) and (5) can be rejected: we do not need to assume that all learnable concepts are definable in terms of the primitive, unlearned concepts of the system, and so the domain of learnable concepts is not restricted in this way by the organism's innate endowment. So unless there are serious objections to the Cognitivist theory that have been overlooked here, it looks like the best theory of language learning in the field.

BIBLIOGRAPHY FOR CHAPTER V

Brachman, R.J. (1979) On the epistemological status of semantic networks. In N.V. Findler, ed., Associative Networks. New York: Academic Press.

Burge, T. (1979) Individualism and the mental. In P.A. French, T.E. Uehling, Jr., and H.K. Wettstein, ed., Midwest Studies in Philosophy, Volume IV, Metaphysics. Minneapolis: University of Minnesota Press.

Evans, G. (1973) The causal theory of names. Reprinted in S.P. Schwartz, ed., Naming, Necessity and Natural Kinds. Ithaca, New York:Cornell University Press, 1977.

Fodor, J.A. (1975) The Language of Thought. New York: Crowell.

Fodor, J.A. (1978) Tom Swift and his procedural grandmother. Cognition, 6, pp.229-247.

Fodor, J.A. (1980) Fixation of belief and concept acquisition. In M. Piatelli-Palmarini, ed., Language and Learning. Cambridge, Massachusetts: Harvard University Press.

Fodor, J.A. (1980a) Reply to Putnam. In M. Piatelli-Palmarini, ed., Language and Learning. Cambridge, Massachusetts: Harvard University Press.

Fodor, J.A. (ms) The current status of the innateness controversy. Forthcoming.

Fodor, J.A. (ms-a) The philosophy of mind. Scientific American, in press.

Fodor, J.A., Garrett, M.F., Walker, E.C.T. and Parkes, C.H. (1980) Against definitions. Cognition, 8, pp.263-367.

Fodor, J.D., Fodor, J.A., and Garrett, M.F. (1975) The psychological unreality of semantic representations. Linguistic Inquiry, 6, pp.515-531.

Forster, K.I. (1979) Levels of processing and the structure of the language processor. In W.E. Cooper and E.C.T. Walker, eds., Sentence Processing. Hillsdale, New Jersey: Lawrence Erlbaum.

Greenberg, B. (1978) Notes on the Progrmming Languaage LISP. Unpublished mimeo distributed at MIT.

Harman, G. (1975) Language, thought and communication. In K. Gunderson, ed., Language, Mind and Knowledge. Minneapolis: University of Minnesota Press.

Katz, J.J. (1966) The Philosophy of Language. New York: Harper and Row.

Katz, J.J. (1972) Semantic Theory. New York: Harper and Row.

Kintsch, W. (1974) The Representation of Meaning in Memory. New York: Wiley.

Loar, B. (ms) Mind and Meaning. Cambridge: Cambridge University Press, forthcoming.

Marslen-Wilson, W. (1973) Speech Shadowing and Speech Perception. Unpublished doctoral thesis, MIT.

Marslen-Wilson, W. and Welsh, A. (1978) Processing interactions and lexical access during word recognition in continuous speech. Cognitive Psychology, 10, pp.29-63.

Miller, G.A. and Johnson-Laird, P.N. (1976) Language and Perception. Cambridge, Massachusetts: Harvard University Press.

Putnam, P. (1975) Mind, Language and Reality. New York: Cambridge University Press.

Putnam, H. (1930) What is innate and why. In M. Piatelli-Palmarini, ed., Language and Learning. Cambridge, Massachusetts: Harvard University Press.

Rosch, E. (1977) Classification of real-world objects. In P.N. Johnson-Laird and P.C. Wason, eds., Thinking: Readings in Cognitive Science. New York: Cambridge University Press.

Winston, P.H. and Horn, B.K. (1981) LISP. Reading, Massachusetts: Addison-Wesley.

Woods, W.A. (1975) What's in a link. In D.G. Bobrow and A. Collins, eds., Representation and Understanding. New York: Academic Press.