THE BEHAVIORAL AND BRAIN SCIENCES (1983) 6, 391–421
Printed in the United States of America

How are grammars represented?

Edward P. Stabler, Jr.
Centre for Cognitive Science, SSC, University of Western Ontario, London, Ont., Canada N6A 5C2

Abstract: Noam Chomsky and other linguists and psychologists have suggested that human linguistic behavior is somehow governed by a mental representation of a transformational grammar. Challenges to this controversial claim have often been met by invoking an explicitly computational perspective: It makes perfect sense to suppose that a grammar could be represented in the memory of a computational device and that this grammar could govern the device's use of a language. This paper urges, however, that the claim that humans are such a device is unsupported and that it seems unlikely that linguists and psychologists really want to claim any such thing. Evidence for the linguists' original claim is drawn from three main sources: the explanation of language comprehension and other linguistic abilities; evidence for formal properties of the rules of the grammar; and the explanation of language acquisition. It is argued in this paper that none of these sources provides support for the view that the grammar governs language processing in something like the way a program governs the operation of a programmed machine. The computational approach, on the contrary, suggests ways in which linguistic abilities can be explained without the attribution of any explicit representation of rules governing linguistic behavior.

Keywords: cognition; computation; grammar; language; learning; linguistics; program; representation; rule-governed behavior

This paper will consider whether there is evidence to support claims that some human behavior is "rule-governed" in something like the way that the behavior of a programmed computer is "rule-governed." In particular, I will examine in detail a claim that in certain cases the behavior of an organism is the result of a computational process that is governed by a set of internally represented rules or formulae of some programming language. There have been many proposals that could 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"), and this sort of claim has since become quite popular. 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 & Arbib 1975). Similar accounts have been adopted by psychologists. Motor-learning theorists talk about the programming of motor behavior (e.g., Stelmach 1978), and psycholinguists have compared natural human languages to higher-level programming languages, which are compiled and then executed (Fodor et al. 1974; Johnson-Laird 1977). I will concentrate, however, on the claim made by Noam Chomsky (1965; 1969; 1972; 1975; 1980a) and other linguists that human linguistic behavior is somehow governed by an "internalized" generative transformational grammar. This proposal has received a good deal of attention, and in the controversy it has engendered the computer analogy has frequently been appealed to. I will argue that there are no grounds for assuming that the grammar or any other rules govern linguistic behavior in the way that a program governs a programmable computer, and in the course of this argument the difficulties facing any psychological theory that makes such a claim will become clear.

The computational framework

Despite the difficulties we may have in grasping or spelling out the details of any particular computational description of a physical system, there should be nothing mysterious about computational description in general. It is 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 that because of their design are conveniently described in computational terms and useful for this reason. But actually any physical system can be given some computational description or other. Let me explain what I mean by this and define some terms that will be used below.

We can provide a computational description of a physical system by specifying a "realization function," a 1–1 mapping or "encoding" (f_R) 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 realization function such that in some specifiable circumstances, C, the system will always, in virture of natural laws that apply to the system, go from one state, s_i, into another, s_f, where for every such pair of states, the symbol associated with the final state, $f_R(s_f)$ is a function F of the symbol associated with the initial state f_R (s_i). In such a case I will say that the system *computes the function F*. This is the basic idea upon which all computational descriptions rest. I will call a theory that provides such a description of a physical system a *first-level* computational theory.[1] I will consider theories that provide

0140-525X/83/030391-31/$06.00

much richer descriptions of physical systems, but any one of them which describes some sort of physical system as computing a function thereby qualifies as a first-level theory.

Since programs are so common in computational theories, 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. Of course, a program that yields a unique value for each different input defines a function, so a program may be used in even a first-level theory to specify the function computed by the system.

Programs sometimes provide more than just a specification of the function computed, however. A program typically expresses an algorithm according to which a set of input symbols are transformed into the corresponding output symbols in a sequence of steps. The symbols that result from the execution of every step of the procedure may be in the range of the interpretation f_R, in which case the system may compute the function specified by the algorithm by passing through each of the intermediate states; it may execute each of the appropriate instructions (I1, I2, . . . , In) of the program by computing the functions (F_{I1}, F_{I2} . . . , F_{In}) specified by the instructions (taking the final state or output of the computation of each instruction I_i as the input to the computation of I_{i+1} for $1 \leq i \leq n - 1$).[2] In such a case I will say that the system *executes* or *computes* the program. It computes the function specified by the program by computing the appropriate sequence of functions. I will call a theory that is committed to the view that a system can be described as computing a program a *second-level* theory.

A second-level theory, then, is necessarily a first-level theory, since it describes a system as computing certain functions. When a second-level theory describes a system as computing a nontrivial program (i.e., a program with more than one step) in the manner just described, it just has more content than a strictly first-level theory, which simply describes the same system as computing the function defined by the program.

Sometimes a computational theory will provide not only a specification of a program that is computed by a certain system, but also some account of how the system actually carries out the computation. In the jargon of the computer engineer, such theories describe a physical system as computing a program "because its operation is governed by an actual representation of that program." We will cash this out in terms of a precise definition of a *program-using* system.

A program-using system is a system that *uses some program P to govern its computation of that program*, where this is explained as follows. Such a system has an encoding of the program itself. We can think of the program as a set of formulae, "instructions," and then the encoding is a (1–1) "program-realization" function f_P, which maps these instructions into states of the system. The system also has a set of states, which we can call "control states," which are associated with the encoded instructions in such a way that the system always computes the function associated with the encoded instruction that is associated with the control state of the system. In other words, the system computes function F corresponding to instruction Ii because it is in a control state which determines that the encoding f_P(Ii) will determine that F_{Ii} is computed. Thus if the control mechanism had been in a control state which made some different encoding f_P(Ij) causally efficacious, then the function F_{Ij} (possibly different from F_{Ii}) would have been computed. In such a case, when the system computes a program P because it is using an encoding of P to govern its operation in this way, the system *uses the program* to govern its operation. A theory that is committed to the view that a system operates in this way is what I will call a *third-level* theory.

A theory that proposes that some system is a "stored-program computer" (in the ordinary sense) will be a third-level theory (in the sense just defined), since conventional computers work in just this way. A third-level theory is also necessarily a second-level theory, since it is committed to the view that a program is computed. A second-level theory that did not provide any account of the mechanism by which a system computes the program would not be a third-level theory. And a second-level theory can fail to be a third-level theory if it provides an account of how the computation is carried out that does not involve the use of an encoding of the program computed. Let me explain this last point carefully, since it will be of some importance.

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. We can, for example, take electronic circuits that compute some very basic functions and assemble them in such a way that the assembly as a whole computes a function that is a particular composition of the basic functions. In this way, networks of electronic circuits can be "hardwired" to compute even very complex programs directly, without having control states to govern their operation according to an encoding of a program.[3] In these systems no explicit encoding of the program is accessed or manipulated. Of course (as follows from the account above), the crucial difference between program-using systems and others is not whether some part or other of the system is "wired in" but whether there is a set of states in the system that encode the program and that have the appropriate causal role in controlling the operation of the system (as specified in the definition of "program-using system" above).

In fact, any program that can be computed by any system can be computed directly, *or* by a program-using system, *or* by some sort of "hybrid" system that uses a program to govern its computations of only certain steps of the program. It is easy to show that simple circuits wired up in the fashion just mentioned can compute any program. It is also important to notice that there is a significant difference between first- and second-level computational descriptions, on the one hand, and third-level computational descriptions, on the other. All three kinds of theory involve giving a symbolic interpretation to relevant states of the system; in the jargon of the computer user, we might say that all three describe certain behavior of the system as the "manipulation of representations." But whereas the first two levels of computational theory use a program simply to describe how the encoded symbols are transformed, we ascend to the third level only when we make a claim about *why* the specified computations are carried out. A direct system transforms representations of the objects of the computation – i.e., it transforms the encoding of the "data" or "memory set" – but a third-level theory has the additional commitment to the existence of both an encoding of the rules or instructions of the program itself and a mechanism that uses these encoded instructions to control the transformation of the memory set. A program-using system is thus distinguished by the way its computational processes are controlled.

Contrasts with other frameworks. It is important to see how first-, second-, and third-level theories as they have been defined here compare with other theoretical approaches. Suppose, for example, that a psychologist proposes a third-level theory: he proposes that under a realization mapping of a certain sort, humans have internally encoded a program P that is used in certain circumstances, say, to process certain sensory input. Such a proposal obviously contrasts with any (strict) behaviorist theory, since it has quite definite commitments to internal states of the organism. And this proposal would also require more than just (the trivially satisfied condition) that the program P be somehow physically encoded in the organism, since it requires that these encodings have (in both actual and possible situations) the appropriate causal roles, as described above (and see note 1). A (strict) behaviorist would presumably also object to any first- or second-level theory that claimed that computations over nonsensory and nonbehavioral states were carried out, although he would probably not object to simple computations from stimulus input to behavioral output.

Our framework could be cast in different and perhaps more familiar computational terms, but then one must take care to avoid letting the terms suggest commitments that theories in the present framework do not (or need not) really have. For example, in the jargon of the computer-using community, the framework proposed here provides an account of when a theory makes claims "above the level of hardware." Obviously, we get above the "hardware level" when we propose third-level theories about the "software," i.e., about the programs used by a system. This terminology is unfortunate, though, insofar as it suggests that the "software" is something other than "hardware," something other than a physical part or feature of the system, or something other than the "wiring." These suggestions of the colloquial terms are incorrect. The present account makes clear the commitment to an actual physical and causally efficacious realization of any program used by any physical system. This realization might well involve wiring, or switch positions, or neurophysiological states, or any number of other undeniably physical features. The important thing is not the "physicalness" of the encoding of the program but the role of the encoding in governing the computational processes of the system.

There are other explicitly computational theories that are not easily translated into the terms we have defined here. For example, some have argued that the notion of an encoding that has some appropriate causal role in the functioning of a system, in one of the senses defined above, does not capture any appropriate idea of the "representation" of a memory set or program. Pylyshyn (1980; forthcoming) takes a position rather like this when he says that we ought to distinguish "representation-governed" processes from the "fixed capacities of mind," which he calls the "functional architecture." This distinction corresponds roughly to our distinction between systems that *use* programs and those that do not, except that Pylyshyn does not explicate his notion of a "representation" in terms of "encodings" with appropriate roles. But, of course, if we do not know what counts as a "representation" in his sense, the empirical content of theories formulated in his terms is unclear.

Another class of theories that is familiar in cognitive science and that suffers from a similar and related difficulty is that of the mentalist theories in psychology, i.e., those theories which are formulated in (inter alia) our pretheoretically familiar mental terms. Of particular interest in cognitive science are theories that explain human behavior in terms of beliefs and desires (both conscious and "tacit" or "unconscious"). There is an ever-growing philosophical literature testifying to the difficulty of understanding what "mental" states are and whether it is appropriate to mention them in scientific explanations of human and animal behavior. Again, insofar as these matters are unclear, the significance of mentalist theories is unclear, as is the relation of these theories to theories expressed in computational terms. [See Dennett: "Intentional Systems in Cognitive Ethology" *BBS* 6 (3) (this issue).]

There is one tradition in this controversy, though, that ties computational and mentalist theories very close together with the assumption that propositional attitudes (like belief and desire) are "computational relations to mental representations." Views of roughly this sort are advocated by Fodor (1975; 1981a), Field (1978), Pylyshyn (1980; forthcoming), and others. On this view and the further assumption that the notion of "computational relation to a mental representation" could be explicated in the vocabulary defined here, any propositional-attitude psychology could be explicated in our terms. But the relations between these different sorts of accounts are matters of current controversy, and the whole story promises to be exceedingly complicated. In any case, the computational terms we will use in the present investigation have the enormous advantage of being relatively precise and unmysterious, so we can be clear where others have not been.

The representational hypothesis

Now let's consider the theory that language users employ an "internalized" transformational grammar when they exercise their linguistic abilities. Do proponents of this theory mean to suggest that the grammar is encoded and used by a program-using system? As we will see, some of their remarks suggest that they do, but it should be kept in mind that we are bringing to bear distinctions that are not usually attended to.

The theory I want to consider has been most clearly formulated by Noam Chomsky. He proposes that each natural language is described by a generative transformational grammar, and he proposes the following hypothesis:

(M) The grammar is mentally represented and used in

> the exercise of linguistic abilities such as understanding speech and making grammaticality judgments.

I will call this the "mental-representation hypothesis" (M). It is usually stated in these mentalist terms, and it has, unsurprisingly, generated some controversy, which I will discuss below. 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 better than any alternative theory does:

> The evidence bearing on the hypothesis attributing rules of grammar to the mind is that . . . facts [about language] 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 assume that the grammar is represented and used in order to explain certain facts about human linguistic abilities. Human language processing is distinguished in this respect from simpler processes that presumably do not use represented rules to govern the behavior. Chomsky points out that things like a person's riding a bicycle (1969, pp. 154–55), the flight of a bird (1975, pp. 222–23), 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.

As was noted above, there is some controversy over the status of mentalist theories. The issue manifests itself in this context particularly as a worry about what "mental representations" are. This worry is often answered with recourse to an explicitly computational perspective. Thus when Harman (1967; 1969) raised issues about the mentalist vocabulary in these hypotheses, Chomsky responded with the following remarks:

> . . . we postulate unconscious knowledge of the rules of the grammar if this postulation is empirically justified by the role it plays in explaining the facts of use and understanding and acquisition of language. . . . I see no objection to saying that "knowledge of these principles" (obviously, unconscious knowledge) is innate, though I do not want to insist on this (in my view, perfectly appropriate and understandable) terminology. It would be easy to program a computer in this fashion. . . .
>
> Consider now language use. I proposed that the mature speaker has internalized a grammar with specific properties that I and many others have discussed in many places, and that in understanding speech he makes use of this grammar to assign a percept a signal. Again, it is possible to design a computer that operates in this manner (and, in fact, there has been a fair amount of experimentation with such programs). There are many possible ways in which such a program might make use of the rules of the stored grammar; it is a central problem of psycholinguistics to explore these possibilities. (1969, pp. 155–56)

The computational perspective endorsed in these passages has not been abandoned by Chomsky or by many of the linguists and psychologists that have taken up this kind of approach. As the passages quoted earlier indicate, Chomsky still endorses the psychological claim that mature speakers have "mentally represented" grammars of their languages, and that these represented grammars are used "in computations eventuating in behavior such as judgements about form and meaning." One can agree with Chomsky's view that this sort of view makes perfect sense and that it is a central problem of psycholinguistics to consider how represented rules of grammar might be used. Psycholinguists also ought to consider whether the attribution of represented rules is needed to explain linguistic behavior, as Chomsky claims. It is remarkable that this sort of investigation has not yet been undertaken with any rigor, either in this context or in any of the other areas in which similar computational theories have been proposed. I propose to remedy this situation, casting the Chomskyan views in the relatively precise terms that I defined above. Those who do not endorse this computational construal and yet want to make similar psychological claims are faced with the difficulty of explaining what a realistic acceptance of their claims commits them to.

There are three main points in the Chomskyan position that suggest that a third-level theory is being proposed. First, it is claimed that the rules of grammar are "mentally represented," and it is suggested that this "representation" is rather like the "representation" of rules in a computer. In the precise terms defined above, we would construe this claim in such a way that it entails at least that the rules are *encoded*. Second, these "represented" rules are assumed to actually govern language processing, not merely to describe it. They are used to "generate" the mental representations used in language understanding, for example. This appears to be closely analogous to the third-level commitment to the view that encoded rules are *used*, rather than simply *computed*. And finally, this sort of system, a system that "follows" "represented rules," is distinguished from simpler systems that do not use them. These points are precisely what a third-level theory would be committed to. It would have a commitment to encoded rules or instructions that control the computation. And whereas virtually any system can be given a second-level computational description according to which "representations" (i.e., an encoded memory set) are transformed, a third-level computational description according to which the system computes a nontrivial program is true only of certain quite complex and structured systems. So perhaps at last these claims about the role of the grammar, which have aroused so much controversy, can be provided with a clear explication in our precise computational terms. Let's make a third-level construal of the claims explicit. We can then assess it and compare it with other proposals about the role of encoded rules of grammar and also with proposals that do not presume that the rules are encoded at all.

The grammar itself is not a language-processing algorithm, of course. But we can assume that the grammar is *used* in the course of executing such an algorithm. We can make this claim precise in the terms defined above. The claim that the grammar governs language processing could be construed as entailing the following sort of second-level 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 linguistic representations.

And so, then, the claim that the grammar is mentally represented and *used* in language processing would be construed as committing us to the following third-level claim:

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

This last proposal clearly commits us to what I am calling a third-level theory. I will argue that, although third-level theories are certainly suggested by the remarks of many linguists and psychologists, and although no other plausible construal of these remarks suggests itself, third-level theories, and this one in particular, are not supported by any available evidence and are probably not something that the linguists or psychologists really want to propose.[4] However, as was pointed out above, one might want to claim that the grammar is not encoded and used as a program but rather is encoded in the memory set "as data" over which other computations are defined. This proposal will also be considered, but, as we will see, it is not so natural a construal of the linguists' suggestions, and it is not supported by available data either.

The grammar as program

My main argument against construing the mental-representation hypothesis as a third-level claim is that the evidence linguists have offered in support of the mental-representation hypothesis (M) does not support the third-level hypothesis (H3). The evidence for the mental-representation hypothesis (M) comes from three related sources in linguistic theory: (1) the explanation of language comprehension and other abilities; (2) evidence for formal properties of rules of the grammar; and (3) the explanation of language acquisition. I will consider each of these sorts of evidence in turn.

Explaining linguistic abilities. Chomsky gives some examples of the sort of evidence that is used to support the mental-representation hypothesis (M) 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 hypotheses like the following:

> 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 principles, the grammar, is assumed to be "one basic element in what is loosely called 'knowledge of language.' " The assumption that the grammar is *used* then explains linguistic abilities, such as the ability to understand a natural language. That is, it explains why the speaker of the language respects the generalizations captured by the grammar in the exercise of his linguistic abilities. Or rather, it explains why the speaker's performance respects the grammar insofar as it does; presumably, other performance factors will also be required to explain aspects of linguistic behavior. In any case, we have an argument here for supposing that the grammar is mentally represented and used in language processing, as the linguists claim.

Granting that this is one of the basic forms of argument for the mental-representation hypothesis, our question is whether it has any plausibility as an argument for the third-level hypothesis (H3). If the argument does provide support for the third-level hypothesis, we should be able to render it in a form that is a little clearer about its computational commitments. So, to begin with, there is perhaps a viable 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. The argument for this claim is already quite clear. It could be put 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 nongenerative 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 *executed* (in the sense defined above) to generate the requisite representations at appropriate points in the computation, and to apply the other appropriate rules and principles to the representations so generated. There are procedures, such as what are called "analysis-by-synthesis algorithms," which use a generative grammar in this way, though these are perhaps only crude first guesses 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 a procedure that has the grammar 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 aspects of the speaker's linguistic behavior accord with the grammar – the procedures computed actually involve the execution and application of the grammatical rules.

This is an argument for the second-level hypothesis (H2), though, and not an argument for the third-level hypothesis (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, 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 would be adequate to 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 our argument for the second-level hypothesis (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 internally encoded 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 requisite representations but also *encoded* and *used* in language processing. Since this encoding and use of the rules is what we are assuming is meant by the proposal that the rules are "represented" and "followed," we have no evidence for M as we are construing it.

Similar points have been made before. For example, John Searle (1980) makes exactly the right point 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)

We are making an analogous point about the difference between the second- and third-level hypotheses: the second-level hypothesis asserts only that the language processing *conforms* to the rules of the grammar, while the third-level hypothesis asserts that the processing conforms to the rules *because* the rules are *encoded* ("represented") and *used* ("followed").[5] It should be emphasized that this criticism does not show any "conceptual problems" for third-level theories; it is just that no good evidence has been presented for them. And if one were to abandon the third-level claims on this ground, one could still maintain a first- or second-level hypothesis about the significance of grammars. One could maintain for example, that in understanding we compute the function from phonetic representation to "logical form" that is defined by the grammar. Such a claim would still have the "realist" commitments to actual encodings of the structural descriptions generated by grammars but would not have any commitment to a causally efficacious encoding of the rules of the grammar. These matters will be considered in detail below.

In any case, if Chomsky *is* proposing H3, then we have done what he says the critic must do, which 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 the speakers do is 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 encoded and used. The basic point is just that some devices compute a function F or a program P without having anything which we could regard as an encoding of a particular name of F or representation of P which is causally responsible for the computation. One sort of alternative account that does not attribute represented rules is the theory that the rules are computed "directly" or by a "hybrid" system, i.e., by a system that is not a program-using system. There will always be possible systems of this sort that are capable of computing whatever program a program-using system can compute. This alternative may in fact be preferable, since direct, hardwired computation is typically much faster and more efficient than computation on a program-using system, and one of the most striking features of human linguistic behavior is its speed and versatility. If, on the other hand, Chomsky and other linguists do *not* mean to propose the third-level hypothesis (H3), then the failure to provide the needed evidence is not in the least surprising. If they are not proposing H3, though, it is not at all clear what they *are* proposing, and we ought to worry about whether we can justify the current emphasis on program-using systems in theories about how people process language. We will return to these points.

The commitments of formal universals. The second line of argument for the mental-representation hypothesis (M) that ought to be considered here 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, i.e., about their actual vocabulary and syntax, 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 the third level (H3).

Fodor et al. (1974) point out this line of argument in a discussion of how linguistic universals ought to be accounted for. They say:

> . . . 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. 369–70)

I think that Fodor et al. have fallen prey to a confusion here about the status of linguistic universals that 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 generative power of the rules. Thus, we can think of them as generalizations that are true of the *computations* or *operations* performed on the linguistic structures posited by the theory. We do not need to think of them as generalizations about the syntax or vocabulary of rules as they are encoded in the human sentence encoding–decoding mechanism or anywhere else. 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 of no significance.

Chomsky makes the relevant point in the following passage from "Conditions on Transformations" (1973):

> 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)

The 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 possible human languages that there must be two such sets of rules whose application must be ordered in this way, we could capture this fact by beginning all and only 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 *do* need to require that certain rules have a certain form (namely, that 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.

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

Let me illustrate this point with another, more likely, example. 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, which is offered as a "condition on form," has here been expressed as a "condition on function"; it requires that grammatical *operations* apply to linguistic representations having a certain structure, not just to those having a certain number of words or a certain meaning, for example, In fact, Chomsky never does express this constraint formally as a "condition on form," but his discussion of it (in Chomsky 1973) 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) NP1, Aux, Vx, NP2 ⇒ NP2, Aux, BE, Vx, by + NP1.[6]

In this case, the form of the rule (under its standard interpretation) indicates to the linguist what structures it can apply to 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 verb, and the third a verb of a particular category. It is no doubt *convenient* to express grammatical rules in some such form, but it is clear that the structure dependence of the grammatical operations does not require us to do so. The structure dependence is, in fact, entirely neutral with regard to the formal properties of the rules.

This example is typical of what are called "constraints on form." Given some standard interpretation of their formalism, these linguistic universals can be captured by constraining the form of the rules in certain ways. We could assume, as Chomsky suggests, that the rules contain symbols that refer to certain kinds of structures, such as 'NP', 'VP', and so on, but it is clear that the point of this assumption is not that our rules must have some particular vocabulary or syntax; it is to rule out *operations* that are not structure-dependent. What we really have is a constraint on the operation of the grammatical rules, a constraint that will of course be reflected in whatever formalism we choose, but not a formal constraint. The proposed linguistic universals apparently all have this character, so, in sum, it is at least not obvious that there are *any* universals that really commit us to rules of a certain form. As Chomsky points out, the question of whether the form of the rules or the application of the rules should be constrained is not an issue that linguists have worried about. For their purposes, i.e. for the purposes of characterizing human languages, either sort of constraint will serve. But in the present context the issue is crucial, and no grounds have been offered in favor of the formal treatment, so we have no argument here for the claim that linguistic rules must be internally encoded.

Language acquisition. Let's turn now to the third basic source of support for the mental-representation hypothesis. This support derives from certain approaches 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 (1965; 1972), Miller (Miller & Chomsky 1963), Katz (1966), and others originally proposed that the way to solve this problem is to assume that language acquisition involves testing hypotheses about the language being spoken; 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 clearly assumes that the possible grammars considered in the selection process *are* represented. So if this theory of language acquisition is well supported, we certainly have support for the mental-representation hypothesis, and we ought to consider whether we also have support for the third-level theory.

The first question, then, is whether this particular theory of language acquisition is well supported by available data. In fact, it has been modified and developed considerably. The problem has been to suggest plausible constraints on the hypothesis-testing procedure such that the correct grammar could be acquired with reasonable efficiency on the basis of the evidence that seems to be available. The acquisition theories that have been proposed more recently sound rather different from the original hypothesis-testing theory. In recent work, for example, Chomsky has proposed:

> 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)

This proposal is not very well developed yet, and it faces some difficult problems (see, e.g., Pinker, in press). But notice that it is not so obvious that this proposal requires that either the core grammar or the acquired grammar be mentally represented. Indeed, at this early stage the theory appears to be particularly amenable to the view that the grammar is not represented. We might assume, for example, that there is a core grammar that is not encoded and controlling processing, but that is essentially "wired in," needing only certain adjustments and additions to yield a full grammatical competence.[7] However, it is premature to speculate about this sort of feature of a theory that is still so underdeveloped.

There really are no well-worked-out computational theories of language acquisition that have firm commitments on this issue; nor do I know of any good argument to the effect that whatever theory is right, it is bound to be one that assumes the grammar to be represented. So again we have no good line of support for either the mental-representation hypothesis or our third-level construal.

We have now completed our review of the basic sorts of linguistic evidence that have been offered in support of the mental-representation hypothesis. It has been argued that this evidence does not support the third-level hypothesis (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 the various claims that are apparently about formal properties of grammatical rules; and finally, there is no good reason to think that it is required for the explanation of language acquisition. In fact, it is beginning to look as though linguists must not have intended to propose the third-level hypothesis at all; they must not have intended the mental-representation hypothesis to commit them to any such view. However, in discussions of speech-recognition models, some linguists have quite explicitly endorsed third-level theories. Halle and Stevens (1962) suggested that "a set of generative rules must be stored within the machine" to generate representations for comparison with the linguistic input in any "automatic speech recognition scheme capable of recognizing any but the most trivial classes of utterances" (p. 157). Miller and Chomsky (1963) adopted the same assumption with regard to human syntax recognition, and almost everyone has followed suit. These proposals have rarely been contrasted with the obvious alternative views about the computing mechanisms that might be involved. However, it is interesting to consider what sorts of evidence *could* be offered specifically in support of the third-level hypothesis.

Plasticity

Let's begin with a consideration of evidence concerning what is probably the most outstanding feature of program-using systems. This feature is what Pylyshyn (1980) has called "plasticity." If the operation of a computing system is governed by an encoding of some program, then relevant changes in that encoding will produce corresponding changes in the operation of the system. Engineers have designed machines, namely programmable computers, that exploit this feature of program-using systems to great advantage. A logic circuit can also be modified to change its operation, but current technology is such that practical considerations overwhelmingly favor stored-program systems for most applications. They are so much more flexible, more "plastic." One hardly ever thinks of anything but a stored-program computer when one wants to execute a nontrivial program. Perhaps the fundamental point about the equivalence of various sorts of systems has been overlooked partly because of an implicit faith in the assumption that the considerations that so overwhelmingly favor stored-program systems for our technology will also constrain natural systems like the brain. This possibility deserves careful consideration.

First of all, let's note one possible pitfall in arguments based on plasticity. Consider some direct, hardwired computer, such as a simple electronic calculator. The operation of this system might seem very plastic, in a sense; it can be made to compute different functions of different arguments in very short order, simply by pressing certain buttons. This is not because the calculator is a program-using system but simply because it is a system whose operation depends on what is encoded in its memory registers and what input it gets. One could imagine more complicated hardwired systems in which the relations between what is encoded in memory and what is computed are more subtle. In all these cases the apparent plasticity does derive (in part) from the variability of encodings that obviously have an influence on what happens, but these encodings do not govern the computation in the way that a program governs a program-using system. The representations are used, we might say, strictly as *data*. Any system that can be correctly described by a second-level theory is of course influenced by representations, but in program-using systems there is a control mechanism that determines what program is to be computed. Any argument from plasticity to a third-level hypothesis must take account of these subtleties.

Suppose that there were some creature who could learn a human language by looking at a transformational grammar of the language for a few minutes.[8] 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. The reason is clear: it is not plausible that a direct system that computed the grammar could grow, or be built, in such a short time. (Of course, this is an empirical assumption.) So we would reject this theory in favor of the view that the grammar is somehow encoded in the creature and that this encoding 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 represented grammar is itself executed by a program-using system or whether it is used strictly as data by some other procedure. That is, the account is neutral between a third-level theory and the other possible alternative views about the mechanisms responsible for the computation. The apparent plasticity of the organism's linguistic processing does not distinguish among these alternatives, though of course other considerations might do so.

Plasticity with respect to a certain input, then, does support the view that the input is encoded and somehow influences the system, even if it does not itself support a third-level theory about the processing involved. For this reason, evidence of plasticity would certainly be of interest in developing a computational account of a natural system. Unfortunately, it is not found in the sorts of human language processing that the grammar might be responsible for. The acquisition of linguistic competence takes more than a few minutes. We can learn a word in a few minutes or less, but this is not the acquisition of information that anyone assumes to be represented in the grammar of the language. We do not find any evidence of plasticity with regard to linguistic competence that would indicate that an encoding of the grammar influences the operation of any sort of human computing system.

In fact, Jerry Fodor (1983) has pointed out that language acquisition is rather like the development of low-level sensory processes in its lack of plasticity: the language development of normal humans exhibits a remarkably reliable characteristic course of development across a wide range of linguistically different environments. Chomsky suggests that, for this reason, what we call "learning" a language might be better understood as the "growth of cognitive structures along on internally directed course under the triggering and partially shaping effect of the environment," analogous to the genetically determined growth of physical organs. Language-recognition processes and low-level sensory processes share a number of other features also. There is some reason to believe that they are all what Fodor calls "modular" processes; i.e., they are fast, "bottom-to-top" computational processes that are relatively autonomous in that they make use of only a very restricted and specific domain of information. These processes are also distinguished by the fact that they seem to be subserved by particular neural structures. Such modular processes are perhaps the ones in which computation is *most* plausibly carried out directly, without the inefficiency of a mediating control mechanism that must access a representation of the procedure to be executed. It is interesting to note that computational theories of low-level sensory processes (such as those of Julesz, 1971, and Marr, 1979, on vision) have explicitly *not* been committed to third-level accounts. Julesz and Marr, for example, have speculated that the processes they have been investigating are implemented by neural networks with a considerable degree of parallelism that compute "cooperative" algorithms.

Parallelism, neurophysiology, and multiple access

Of course, the mere existence of parallel processing does not in itself show that a system does not use a program. In fact, many of the familiar programming languages allow the programmer to specify steps that can be executed in parallel. This parallelism is usually "simulated" on a machine that is essentially like a standard sequential machine, but this need not be the case. Programmable parallel machines are being developed and will probably become more common in the future. Again, the crucial requirement for a machine's being a program-using system is that its computational processes, whether parallel or not, be controlled by an encoded program.

It is perhaps possible, at least in principle, that neurophysiological evidence could support a third-level theory. No such evidence has been found, however, either for linguistic processing or any other cognitive process. The neurophysiologist David Hubel remarked recently:

> 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)

The lack of theories in neurophysiology that are committed to program-using architectures is really not surprising, and it is probably a mistake to start assuming, on these grounds, that there are no such architectures. The distinction between a program-using system and other sorts of system is really one of detail at the level of the wiring. The computational account of cognitive processes would need to be very well developed before neurophysiological data could be brought to bear on any such issue. The theories of low-level visual processing developed by Marr and his associates are quite sophisticated; yet the underlying neurophysiological mechanisms are only beginning to be understood. And low-level visual processing hardly exhausts the field of visual perception, let alone the field of cognitive processes. In the area of language processing the underlying neurophysiology is not well understood and cannot as yet tell us anything about the issues we have been considering here.

Let's consider some of the other sorts of evidence that might be considered relevant. Any chronometric or other complexity-related result that can be accounted for by a third-level theory can be accounted for by a corresponding second-level account and an assumption of "direct" computation. It has been suggested that we should assume that the grammar is mentally represented since it embodies information that is used in many different sorts of activity. It might seem that rather than having, say, language-comprehension algorithms that simply compute the functions defined by the grammar and linguistic-judgment algorithms that also happen to compute related functions, it would be more reasonable to make the assumption of modularity with respect to the grammar, to assume that it is represented in one place and accessed by other cognitive systems. Then one could also make sense of the possibility that a person could have knowledge of his language and yet be unable to use it; as Chomsky (1980a, pp. 51–52) points out, this is a possibility we want to allow. But this kind of argument (which Pylyshyn, 1980, aptly calls the "multiple-access" argument) does not support any third-level claim. We could have a "modular" linguistic processor that is a very fast, direct computing system that computes the functions from one level of linguistic representation to another, and this "module" could be used in the exercise of any linguistic ability. It could be used in the course of understanding language *and* in deciding whether a string is grammatical. To make such decisions we might, for example, simply use this module to check (unconsciously) to see whether the string can be assigned a well-formed linguistic structure. It is clear that we could perfectly well have such "multiple access" to systems that do not use an encoding of the rules they compute. There is no reason to think that any activity involving linguistic abilities would need any more than this.

Summary. Returning to our attempt to construe the mental-representation hypothesis (M) as the third-level theory (H3), it is remarkable that linguists typically do not mentiòn plasticity. Plasticity is what program-using systems are famous for. If a third-level construal were intended by the advocates of the mental-representation hypothesis, one would expect a discussion of the absence of this feature. Again, it appears that a third-level theory must not be an appropriate construal: not only does the evidence offered in support of M fail to support H3, but evidence that clearly would be relevant is not considered. We can add to this the observation that many of the linguists' remarks indicate that they do not have anything like H3 in mind. For example, Chomsky compares M to the hypothesis that a missile is controlled by a computer with a representation of a physical theory, without suggesting that the physical theory is embedded in the program computed by the computer (1980b, p. 11). So the natural question to ask at this point is: If proponents of M do not have H3 in mind, then what are they proposing? This question is not easy to answer. As was noted at the beginning of this paper, a number of points indicated that a third-level account was intended; any other construal will need to be reconciled with these points.

The grammar as data

Before considering some alternative construals of M, let's quickly review what has been done so far. We have surveyed the arguments offered by linguists in support of M, the hypothesis that the grammar is mentally represented and used in language understanding. It was argued that this evidence does not support the third-level hypothesis:

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

The problem is that evidence offered in support of M does not serve to distinguish H3 from:

(Hh) In human language understanding, a program P that 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, Hh seems at least as plausible as H3, if not more so. We discovered no evidence for M that supported H3 any more than it supported Hh. So, we concluded, no grounds have been provided for assuming any such program P is computed by a program-using system rather than a system of some other kind.

So suppose that proponents of M do not mean to propose H3. What else might they mean? Well, they might be construed as intending:

(Hd) In human language understanding, a program P′ is computed (either by a program-using system or by some other kind of system) that uses G *as data*.

This hypothesis claims that, under some interpretation, the language-processing system has a representation of G in memory to which certain computational processes are sensitive; parts of G are taken as arguments to functions that are computed. But proponents of M are never so explicit about the role of the grammar, so it is really more plausible to assume that they are proposing that either H3 or Hd is correct. That is, the more likely 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 *somehow*. We can formulate this view as follows:

(Hb) In human language understanding, the program computed either includes G and is computed by a program-using system *or* uses G as data *or* 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 M supports Hb. In particular, does the evidence for M serve to support Hb as opposed to the nonrepresentational hypothesis Hh? We did not consider the looser hypothesis Hb, but a quick review of our discussion suffices, I think, to show that it will fare no better than H3 did. The strategy used against the third-level hypothesis (H3) was to argue that in the light of the evidence, Hh is just as plausible, so there is no reason to favor the third-level theory. If we succeeded in this argument, then it is easy to see that the looser hypothesis Hb is also undermined; the point is that the evidence supports the nonrepresentational models just as well as it supports the representational models. So there is no reason to favor the loose representational hypothesis over the nonrepresentational Hh.

Even if Hb were empirically supportable, it would still have trouble as a construal of the linguists' position. We noted above that three basic points suggest a third-level construal of the mental-representation hypothesis: first, rules of grammar are assumed to be "represented"; second, these represented rules are computed in the exercise of linguistic abilities; and finally, linguistic behavior is "rule-governed" rather than simply "in accordance with the rules." Hb is "looser" than the original third-level construal H3 in that it allows that the represented grammar may be used only as data in the computation of some program. But if this were the case, it is not clear that there would be any grounds for saying that the rules of the grammar were computed or executed to generate representations. Yet it is possible that the procedure that took the rules as arguments might do something to that effect. There would still be a problem, however, with saying that on this account linguistic behavior is rule-governed. It is true that the rules of the grammar would influence linguistic processing, but this sort of influence cannot be a sufficient condition for a system's being rule-governed. If it were, we could show, say, the behavior of the Sun–Mars orbital system to be rule-governed by showing that under some interpretation of f_R the system encodes rules that are taken as arguments in the computation of some function F – a nearly trivial exercise. So the hypothesis Hb does not seem to be as good a construal of the linguists' claim as H3 is; and, as we have seen, even if this were what they meant, the hypothesis is as unsupported by available evidence as H3. So we are left without any plausible construal of the mental-representation hypothesis. All the reasonable construals have failed us.

Conclusions

As noted above, given the lack of evidence for a third-level theory of grammar one might want to retreat to a weaker and more defensible first- or second-level hypothesis to the effect that the grammar serves only to define certain functions that are computed in language comprehension, where these are simply the functions that map one level of linguistic representation into another. Berwick and Weinberg (1983a) explicitly adopt this weak first-level hypothesis, for example. However, as we have seen, such a hypothesis does not warrant the view that the grammar is *represented*, even in the already very weak sense of there being an encoding or realization function f_R or f_P that associates the rules of grammar with causally efficacious states of the system. Certainly more than a first-level hypothesis has been intended by the claims that the grammar is "embedded" as a distinct component in the recognition system, that it is "mentally represented" and used, that its rules are computed in language comprehension, and that it represents our knowledge of language. It is the latter claims that have aroused such controversy. Few would have been so concerned about the first-level hypothesis by itself. But perhaps, after all, that is the only one that should be taken seriously. As Berwick and Weinberg point out, even if the role of the grammar were only to define the functions that an adequate recognition device must compute, having something to play this role alone is enormously valuable.

So we have reached our rather pessimistic conclusion. Given any program, there are indefinitely many different kinds of computing systems that can compute it. 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 of speculation what sort of procedure is computed (if a computational account is going to be successful here at all). So we have here a positive result of some importance: theories of human language processing that are really concerned with how the language processor is implemented ought to consider the whole range of possibilities. There is apparently no good reason to focus on program-using systems, or on systems that have the processing algorithm encoded in them. The issues invovled in deciding among the various computational accounts that do or do not suppose that a grammar is represented are really quite complex. It appears that we will need a much firmer grasp on what is going on before this sort of issue can be decided.

Marr and Poggio (1976) have suggested that we must have a first-level hypothesis about what function is being computed before we can propose any reasonable hypothesis about what algorithm is computed, and we must have a reasonable hypothesis about what algorithm is computed before we can begin to investigate the mechanisms responsible for the computation. I think that this agenda is overly rigid, but we can conclude, at least, that the firmly entrenched idea that grammars (or any language-processing procedures, for that matter) are "mentally represented" and "used" is in need of defense. On a more positive note, the burden of our argument has been to suggest that even if the representational hypotheses were removed from current theories, very little of the substantial and interesting work that has been done in linguistics and psychology is going to topple as a result. A true (or approximately true) linguistic theory, a characterization of human languages, will not specify the human language-processing algorithm, nor will it provide any indication of whether that processing algorithm is encoded and used in the exercise of linguistic abilities.

ACKNOWLEDGMENTS

This paper is a revised version of part of my doctoral thesis (Stabler 1981). I have really had a lot of help. I would like to thank my thesis readers, Jerry Fodor and James Higginbotham, and also Robert Berwick, Noam Chomsky, William Demopoulos, Howard Lasnik, Justin Leiber, Jay Keyser, Robert Matthews, Zenon Pylyshyn, Amy Weinberg, and Kenneth Wexler for helpful discussions of this material. A version of this paper was read at Brown University on April 7, 1981, at the University of Chicago on February 25, 1982, and at the University of Western Ontario at the eighth annual meeting of the Society for Philosophy and Psychology on May 14, 1982. The discussions of the paper on these occasions were also very helpful. The suggestions of Stevan Harnad and anonymous referees inspired some final and important improvements.

NOTES

[1] Setting up the framework this way makes computational descriptions relatively unmysterious, but of course it says nothing substantial about the interesting question of which systems, exactly, are the ones that are "conveniently" described in computational terms; it does not explain what a "computer," in the usual sense of that term, is. A related problem is that of explaining how to block what Lycan (1981) has called the "fortuitous" satisfaction of the conditions on a computational account. The problem is that if we allow the realization mapping f_R to be sufficiently baroque, any system can realize any com-

putation. Thus, for example, the claim that humans compute internal representations of their linguistic input in a certain way would be trivially true of *any* physical system under some realization mapping or other. This would certainly not be a happy construal of the computational claims. The way to avoid this is not to read computational psychologists as advancing claims of the form "Under some f_R, such-and-such is computed" but rather to recognize that they impose quite definite constraints on the realization mapping. In the case of the processing of acoustic linguistic input, for example, we have fairly clear ideas about what the inputs (which will be in the domain of f_R) must be: they must be states that are regularly caused by the appropriate acoustic events; the consequent computational process must also be a natural causal chain that follows such stimulation in the appropriate cases; and this process must be implicated in the etiology of behaviors that depend upon an understanding of the acoustic input. It is possible that these "causal conditions" will not sufficiently constrain the mapping f_R, but they certainly are a substantial start.

2. Obviously, this is a simplified account. In programming languages that are actually used by computers, the correspondence between the well-formed formulae of a program and the functions to be computed is quite complex, even for the simplest languages. This sort of account of programming languages is provided by "denotational semantics" – semantic theories that interpret symbols of the language as denoting (inter alia) functions over sets that include the memory set of the machine. (See, e.g., Tennant 1976; Stoy 1977.)

3. In fact, there are results comparing the relative complexity of the computations of functions by combinational networks of logic circuits (which do not contain anything that could be regarded as an encoding of the algorithm computed which controls the computation) and Turing machines (whose operation may be controlled by an encoding of the algorithm to be computed on a one-dimensional tape or other storage structure). (See, e.g., Schnorr 1976; Pippenger & Fischer 1979).

4. This undertaking should be distinguished from other, more ambitious projects. For example, one might challenge both (M) and (H3) on the ground that there is another theory that does not presume that a generative transformational grammar is used in 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. (See, e.g., Fodor et al. 1974, ch. 6; Kaplan & Bresnan, in press a.) 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 to support the third-level claim that mental representations of the postulated procedures are actually in control of the processing. However, in this paper I will only consider whether what we have called the "mental-representation hypothesis" ought to be construed as committing us to the third-level hypothesis (H3).

5. I suspect that Searle would want to criticize not only the third-level theory that the rules are represented and used but also the strongly realist second-level theory that they are computed. However, in this paper we are considering only the former claim. A critical examination of the methodology for confirming second-level theories like H2 is beyond the scope of this paper.

6. Chomsky and many other linguists have since come to doubt that passive forms are transformationally derived, but the rule still serves perfectly well here as an example of how structure dependence is reflected in the form of the transformational rules.

7. The view that the core grammar might be built into a language-processing system that computes linguistic procedures directly conforms well, I think, with Jerry Fodor's (1983) proposal that language-recognition processes and low-level sensory processes are "modular." This proposal is further discussed below.

8. This example was suggested to me by Noam Chomsky (personal communication).

Open Peer Commentary

Commentaries submitted by the qualified professional readership of this journal will be considered for publication in a later issue as Continuing Commentary on this article. Integrative overviews and syntheses are especially encouraged.

Using what you know: A computer-science perspective

Robert C. Berwick

Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Mass. 02139

Stabler claims that one cannot take linguistic theory as a true description of the actual (presumably neural) mechanisms that underlie linguistic behavior (language acquisition and use). The reason: lingusitic theory is committed to a program-using model, and the available empirical evidence is equally compatible with a nonrepresentational "hardwired" model. Both prongs of the argument seem flawed, however, for there is evidence that the rules and representations posited by grammars are causally engaged in the machinery of language, and there isn't as big a difference between program-using systems and hardwired systems as Stabler would like to think.

Stabler dismisses three possible domains of evidence that could reveal grammars to be mentally represented and used: linguistic behavior, linguistic universals, and language acquisition; interestingly, contra Stabler's analysis, there is reason to believe that each of these domains does support the hypothesis of mental representation and use.

1. The use of rules. Once we abandon the mid-1960s structural-description/structural-change view of transformational rules, there is evidence that the single "rule" of modern transformational grammar – a rule that takes a syntactic constituent like a noun phrase and moves it – could be causally engaged in language processing. Let me mention just one piece of evidence for this. Recent research carried out by myself and Amy Weinberg (Berwick & Weinberg 1983b) shows that one can explain certain facts about linguistic behavior – in particular, the constraint of subjacency and the observable fact of efficient sentence processing – when these rules are quite literally embedded and causally engaged in a model of language processing. Crucially, the explanation does not hold when one adopts alternative rule systems that attempt to describe the same surface facts by something other than a movement rule. That is, alternative conceptions that obey the same generalizations (hence conform to the same rules and principles as the transformational grammar, in Stabler's terminology) but don't use the same mechanisms fail to explain why a constraint like subjacency should exist. This, then, is a genuine case where a particular grammatical rule supports a particular claim about what sort of mechanism is responsible for a (mental) computation – just what Stabler has said would be required to show that grammatical rules are "used."

2. Half a loaf is better than none. There's good evidence that the data structures – the units of representation – posited by theories of transformational grammar are actually engaged – causally implicated – in on-line language processing. This, after all, was the burden of Fodor, Bever, and Garrett's (1974) argument. Algorithms being data structures plus program control, it would be truly bizarre for the data structures associated with a certain program to be causally engaged in actual computation, but not the associated program. Admittedly, it isn't necessary that this be so, but it certainly constitutes a prima facie case. Of course, Stabler will insist that this doesn't force us to adopt a causally engaged grammar, but it seems to me that the burden of proof goes the other way. See also the discussion on compilation below.

3. Compilation and acquisition. Suppose there was a device whereby you could write a program that carries out the computation of some function – like, say, a fast fourier transform – and presto! out pops a piece of silicon that actually carries out that computation. Now here's the $64K question: If you now use this new circuit, is the original program you specified causally engaged in the on-line computation of the function? Answer: well, no, since the thing isn't plastic. There is no sense in which individual instructions are looked up and their corresponding actions carried out, because there is no control mechanism at all; it's just a single circuit. So the chip seems to meet all the tests for a *non*-program-using system, at least by Stabler's own account. But was your original program used by the chip-making machine? I think that anyone would say yes – yes, the program was used to direct the production of the chip, and it was used, not as some simple kind of "data," but in a basic and fundamental way as the controlling program for the creation of a new artifact. The analogy to language acquisition should be plain: modern transformational grammar is expressly designed as a causal explanation of the "construction" of language competence. (For a worked-out account of the parameter-setting model that Stabler insists will have "problems" – incidentally illustrating a rather richer sense of a grammar being indirectly encoded in a processing model – see Berwick 1982.) Stabler tries to dodge this possibility by talking about this kind of indirect causal connection as merely "influencing" linguistic processing, but the kind of "influence" we are talking about here is precisely one in which the rules and principles of the grammar are actually used (in the ordinary sense of the word).

Finally, let me point out that the possibility of compilation muddies the program-using and hardwired waters to such an extent that the distinction becomes worthless. Stabler allows that a rule is "used" if it is causally efficacious (under some encoding) in actual linguistic behavior, pointing out that one could have a "hardwired" circuit that simply computes a function without "anything which we could regard as . . . a . . . representation of [a program] P." But the three little words "under some encoding" make all the difference in the world. The technical results on the interchangeability of circuits and programs that Stabler cites in his note 3 actually assume that there is a (uniform, space- or time-bounded) encoding of such "hardware" circuits in terms of some program's operation. That is, given any such circuit, there is an equivalent program that simulates, step by step, the algorithm instantiated by the circuit. But this means that the hardwired circuit *is* a program-using system under Stabler's formal definition, since it *is* using an encoding of a program P, namely, the one provided by the (uniform) mapping between circuits and programs (actually, families of circuits). To be sure, the encoding is complex and implicit in the network, but so much the worse for the formal definition. (There is an attempt to find an escape hatch to this difficulty later on, when Stabler strengthens his definition to be an "explicit encoding," but since no explanation of "explicit" is provided, we're left in the same fix.)

We can make the same point another way by seeing whether such a circuit meets another of Stabler's tests for true program-using behavior, namely, "whether there is a set of states in the system that encode the program and that have the appropriate causal role in controlling the operation of the system." The answer is yes, since to answer why computations are carried out, we appeal to the associated instructions of the (encoded) program. But surely, one might counter, the program statements aren't really there, and so one needn't appeal to a program. Doesn't the circuit just "do it" in virture of its (physical) construction, without the causal intervention of program statements (however complicated)? Well, yes, in a sense – but then, if we allow that as the basis for why a computation is carried out, the same holds true for a machine executing the compiled instructions of a higher-level programming language. After all, circuits are what direct the operation of the machine at the bottom, not the program at all. (Remember that after it is compiled, the explicit instructions of a higher-level programming language aren't there at all, only "some encoding" of them.) But we would still say that the program is *used* by the machine, even the compiled program. One can't have things both ways – either both circuit and higher-level languages are program-using systems, or neither are. And talking about "control states" won't help us either. Control states, after all, don't really exist, except under our attribution of them – the attribution used to explain (and direct) the behavior of the machine. In fact, there aren't any states at all, except insofar as we (as intelligent observers) attribute them, and this is true even of program-using systems.

In the end people behave as if they use rules of grammar, even if they don't. And linguistic theories seem to give us true descriptions of the world. What more could one ask for? Now, perhaps this is just an artifact of human theory-making abilities: because we can design and think about program-using systems, and because there is a way to map from hardwired systems to program-using systems, we build program-using-type theories. From another point of view, though, this is exactly why I don't share Stabler's fear that "we ought to worry about whether we can justify the current emphasis on program-using systems in theories about how people process language." It's the only game in town.

On a computational perspective without substance

Rudolf P. Botha

Department of General Linguistics, University of Stellenbosch, Stellenbosch 7600, South Africa

Stabler's conclusion that the basic sorts of linguistic evidence that have been offered in support of the mental-representation hypothesis (M) do not support the third-level hypothesis (H3) is, as he suspects, "not in the least surprising." For, at least within a strict Chomskyan approach, H3 cannot be a good construal of (M). Thus, a finding to the effect that H3 is supported by the evidence for M would have been quite remarkable.

For H3 to be an accurate reconstruction of M, it must be possible to interpret Chomsky's use of the computer analogy as a deliberate attempt at clarifying the ontological interpretation of grammars with the aid of computational notions (for a recent use of this analogy see Chomsky 1980a, p. 188). And such a construal of the computer analogy is quite problematic.

Had Chomsky seriously "endorsed the computational perspective," to use Stabler's terminology, it would be reasonable to expect this "perspective" to affect the linguistic analyses typically performed by Chomsky and his close followers. As argued by Botha (1980; 1982), however, Chomsky's use of the computer analogy and of expressions such as "(mental) computation" has no substantive consequences for these analyses. Thus, Chomsky's "endorsing of the computational perspective" does not contribute in any way to the empirical content of linguistic claims such as those found, for example, in his *Lectures on Government and Binding:* neither language-independent claims (e.g., those embodied in subjacency or the binding theory) nor language-specific claims (e.g., those embodied in the categorial rules of English) reflect this "perspective" in their empirical content. Moreover, in justifying such linguistic claims, Chomsky and others do not find it necessary to appeal to special evidence of a computational sort. Finally, Stabler's "computational perspective" does not affect Chomskyan linguistic analysis heuristically by generating new grammatical or general-linguistic questions.

Scholars who believe that Chomsky has seriously "endorsed the computational perspective," then, have to face the following question: How is it possible for Chomsky and his associates to perform nuts-and-bolts linguistic analyses without paying attention to the sort of computational considerations pertinent to Stabler's discussion? This question does not receive sufficient attention in Stabler's paper. Botha (1980; 1982) has argued that Chomskyans can make the above-mentioned analyses because Stabler's "computational perspective" does not constitute a substantive element of their approach to linguistic inquiry.

To conclude: the importance of Stabler's paper is not located in the conclusion that the evidence furnished in support of M fails to justify H3. The importance of this paper lies in the more general points that it contributes to the discussion of how Chomskyan linguistic theories may (not) be interpreted ontologically. I briefly mention two of these. First, the paper furnishes yet another clear illustration of the nonilluminating nature of the computer analogy. Second, Stabler's discussion has considerably raised the standards of precision for future attempts to interpret Chomskyan linguistic theories in terms of computational notions.

Church's thesis and representation of grammars

Martin Davis

Courant Institute of Mathematical Sciences, New York University, New York, N.Y. 10012

Stabler criticizes Chomsky's hypothesis that grammars function in certain aspects of language use in a manner like that of stored programs in a computer by placing this hypothesis in the context of a proposed model of computation. He suggests that such a model has not been hitherto available and claims that therefore he "can be clear where others have not been." I will argue that in developing his ideas Chomsky had at hand a profound and highly successful theory of computation, that in the framework of this theory, Chomsky's ideas are entirely natural, and finally that Stabler's model betrays serious confusion about the way in which programs are actually used by computers.

The possibility of giving a precise characterization of the class of mathematical processes that can be carried out by purely mechanical means first came to light during the 1930s in the work of the logicians Church, Gödel, Post, and Turing. (The basic papers will be found in Davis 1965.) Each of these logicians introduced one or more notions as proposed characterizations of this class, and all of these proposals subsequently turned out to be equivalent, in the sense that although the notions themselves were different (in some of the cases very different), the classes demarcated were identical to one another. It was Church who first asserted that all mechanical processes are encompassed by these notions, and the assertion is therefore known as Church's thesis. However, the most compelling evidence for its correctness was supplied by Turing, who was led to his machine model by considering a description of a human being carrying out a calculation and successively peeling away irrelevant details. The work of these pioneers has not only had important consequences in mathematics but has also presaged many important aspects of computational practice which are now commonplace and whose intellectual antecedents are typically unknown to users. Thus Turing proved the existence in principle of all-purpose (or universal) digital computers and showed that hardware can always be translated into software and that programs can be constructed whose function is to "interpret" other programs. Post introduced the concept of a program as a list of instructions and provided the basic concepts needed to represent grammars by formal structures based on productions. (For further discussion of some of these matters, see, for example, Davis 1978. A thoroughgoing analysis of the implications of Church's thesis for mechanism and mentalism can be found in Webb 1980.)

For our present purpose, it is important to emphasize that the infinite size of the domains of definition of the functions contemplated under Church's thesis is no obstacle to a mechanistic interpretation. The grade school algorithm for adding two natural numbers, given as strings of decimal digits, is most naturally stated with no bound on the arguments, although as a practical matter, there is of course a limit on their size. In the same way, the most natural description of the class of grammatically correct sentences of a natural language involves no a priori upper bounds and is readily placed in the context of computability theory as simply being a particular recursively enumerable set of character strings. For Chomsky, Post's production rules provided the perfect bridge: they could be used, on the one hand, to generate arbitrary recursively enumerable sets, and on the other, to model rules of grammar.

Now we are ready to consider Stabler's argument that "there are no grounds for assuming that the grammar . . . govern[s] linguistic behavior in the way that a program governs a programmable computer." This argument is based on his three-layered account of program-driven computation. There are a number of difficulties with this formulation. Thus Stabler himself points out that "if we allow the realization mapping f_R to be sufficiently baroque, any system can realize any computation." Even more serious is the fact that Stabler's "third-level" account of the way a computer's behavior is governed by a program bears little resemblance to the way actual computers work. The separate instructions are stored in a computer in encoded form like so much data, but they do not in any relevant sense correspond to individual "states" of the computer. Just how a program will be carried out will depend on the language in which it is written and the implementation of that language (via interpreter or compiler or some hybrid) for the computer in question. The distinction between a program being present as "data" and a program "controlling" the operation of a computer is just not clear-cut, and it is not particularly helpful in understanding the behavior of program-driven computers. A programmer who "runs" a LISP program can quite appropriately say that his program is driving the computer. And this situation will obtain regardless of whether the program is being run on a LISP machine or via an interpreter on a conventional computer. In the latter case the interpreter uses the program as "data" which it "consults" in order to "know" what task to undertake next.

This is not the place to repeat Chomsky's arguments for the existence of innate biological mechanisms for learning and computationally applying grammars. Such mechanisms would presumably use an internal representation of a particular grammar and might well operate on such a representation very much in the manner that an interpreter or a compiler operates on a computer program. In such a case it would be just as appropriate to speak of a person's linguistic behavior as being grammar-driven as to speak of a computer as being program-driven.

On the hypothesis that grammars are mentally represented

William Demopoulos and Robert J. Matthews

Department of Philosophy and Centre for Cognitive Science, University of Western Ontario, London, Ont., Canada N6A 5C2

> "Well now, would you like to hear of a race-course, that most people fancy they can get to the end of in two or three steps, while it *really* consists of an infinite number of distances, each one longer than the previous one?"
>
> Lewis Carroll, *What the Tortoise Said to Achilles*

1. Introduction. It is a truism, and so presumably true, that speakers use their knowledge of language in the exercise of their linguistic abilities. Drawing on the characterization of linguistic knowledge made available by modern linguistic theory, proponents of the so-called representational theory of mind propose the following construal of this truism: learning a language is a process that eventuates in the mental representation of a grammar, which is then used by the speaker in the exercise of his linguistic abilities. The assumption that a mental representation of the grammar is so used is said to be warranted by the fact that this assumption, when taken together with independently motivated theories of the character of other interacting variables (such as memory limitations and the like), yields the best explanation of the data about the speaker's mental states and processes or the behaviors in which such processes result (Chomsky 1980a; Fodor 1981a; 1981b). This form of argument is one that any good realist should accept, but it leaves unspecified the intended interpretation of what Stabler calls the mental-representation hypothesis (M). Taking his cue from repeated assertions that the mentally represented grammar is used in computations that eventuate in the behavior to be explained, Stabler considers the possibility that the intended interpretation of M is given by H3, which entails but is not entailed by H2. Examining evidence alleged to support this construal of M, he concludes that at best it supports H2 alone. His argument takes the following form: for any program-using device that computes a function F or a program P, there exists another device that computes that same function or program without having anything that could be regarded as an encoding of a particular name of F or representation of P that is causally responsible for the computation of F or P. Thus, evidence for H3 must support not simply the claim that language understanding involves the computation of some function or program but also the further claim that a particular name of F or representation of P is causally responsible for the computation. The evidence adduced in support of M, Stabler argues, fails to support this further claim; indeed, modularity considerations would seem to favor H2. From this Stabler concludes that H3 is presumably not the intended interpretation of M. But, he asks, "if proponents of M do not have H3 in mind, then what are they proposing?"

Stabler's charitable conclusion notwithstanding, various linguists, philosophers, and psychologists seem to have intended just the computational construal of M that he considers and then shows to be evidentially unsupported. Bresnan and Kaplan (1982b), for example, go so far as to propose H3 as an adequacy condition on linguistic theory. Stabler's critical examination is salutary, for it calls attention to the persistent tendency of representationalists to identify mentalistic psychology with computational psychology. This tendency seems both empirically and methodologically unmotivated. We suspect that the attractiveness of a level-three construal of M stems from the sort of concern expressed by Stabler at the conclusion of his paper, namely, that there seem to be no other reasonable interpretations of M. In the remarks that follow, we sketch what we believe to be a plausible alternative interpretation, one that is supported by the evidence typically adduced in support of M but that differs from H3 in its commitments to particular computational realizations of mentally represented grammars. We conclude with some brief remarks regarding Stabler's characterization of representation-using systems.

2. The interpretation of the mental-representation hypothesis. First let us recall why an alternative to a level-three interpretation is desirable. On a level-three construal of the mental-representation hypothesis, there is a clear sense in which the grammar is encoded in a program that controls linguistic behavior. On this construal a grammatical rule acts causally via its encoding by a sentence in the language of thought. We may therefore conceptualize knowledge of the rule as an extension of the computational model of ordinary propositional knowledge: in both cases we stand in a yet-to-be-specified computational relation to an encoding of the proposition that we would be said to know. As already indicated, the difficulty with the extension of this model to the case of grammatical rules is that there is little reason to believe that this correctly characterizes our knowledge of such rules: the most prominent feature of such a model, namely, its plasticity, seems to be lacking, and many current theoretical and empirical indicators point in the opposite direction (cf. Fodor 1983).

Can we be satisfied with a level-two construal? It seems that we can, but only if we are willing to grant that there is no interesting sense in which the grammar is *known*. Recall that on a level-two construal, all that we are committed to is that the mind computes the functions which correspond to rules of the grammar. On this view, the sense of the claim that the rules are represented and used is exhausted by the (much weaker) assertion that our linguistic behavior *accords with* the grammar; on a level-two construal, our knowledge of grammar is not unlike the "knowledge" that the earth–moon system has of its Hamiltonian, and this seems clearly unacceptable.

So, what we require is an interpretation of the mental-representation hypothesis under which the thesis that grammatical rules are represented justifies the inference that they are used and not merely conformed to. First of all, it is necessary to be clear concerning the domain over which we wish to preserve this distinction. It seems clear that we wish to preserve it over the domain of overt linguistic behavior, that is, we want to say that the computation of semantic representations from phonetic representations (and vice versa) is mediated by knowledge of the grammar, and that the phonetic-representation/semantic-representation pairing does not merely *conform to* the grammar. A representationalist theory achieves this by postulating syntactic representations (i.e., structural descriptions) that are generated and transformed according to the rules of the grammar and that mediate between phonetic and semantic representations. That is, grammatical rules specify functions defined over a domain of internal states; these states are specified in a theoretically autonomous vocabulary, and the internal processes thus defined mediate the production of linguistic behavior.

Does the mental-representation hypothesis have to assert *more* than this? Notice that this construal preserves the idea that our linguistic behavior arises *because* of the grammar we have internalized, and it does so in the straightforward sense of being mediated by internal processes, described in the vocabulary of the grammatical theory, and constrained by the rules of the grammar. But on this construal the internal processes are only asserted to occur *in accordance with* (or to *conform to*) the grammar. This differs from a level-three hypothesis as follows: on a level-three construal our linguistic practice is rule-following and not merely rule-conforming only if rules are followed "all the way in." Thus on a level-three construal the internal manipulation of structural descriptions must also be rule-following. This is the import of the requirement that there be an explicit encoding of the grammar. But our account makes no such demand. Rather, we require two things. First, we require an appeal to the manipulation of internal representations that accords with (or conforms to) the rules of the grammar. Second, we require that within the context of the relevant linguistic theory, and under the normal methodological constraints on scientific explanation, this appeal should be theoretically indispensable for the explanation of linguistic behavior.

One final comment before turning to the evidence for M under this interpretation. It might be urged that on this account we are still left with the problem of distinguishing between hypotheses that appeal to internal processes which are *rule*-conforming and hypotheses which appeal to *law*-conforming internal processes. On the present proposal a process like the internal molecular motion of a gas is a *law*-conforming (rather than rule-conforming) process because the regularity it instances is defined over states characterized in the vocabulary of physics, and presumably there is some interesting sense in which this is *not* true of structural descriptions. So our account leaves this issue resting squarely on our ability to make out the distinction between special-science properties and laws that *are* reducible to properties and laws of physics and those that are *not* so reducible. And this seems exactly where the issues should rest. (On this point, compare Pylyshyn's cognitive-process/functional-architecture distinction in Pylyshyn 1980.)

3. Evidence for the hypothesis. Our construal of M finds support in the sort of evidence that is typically adduced in support of M: such evidence, if taken at face value, would provide support for the claim that a broad domain of linguistic behavior can be explained by appeal to such grammatically characterized internal processes. To the extent that we lack alternative explanations of these same behavioral phenomena, this evidence provides support for the theoretical indispensability of such appeals. On our interpretation of M, evidence of the sort typically adduced provides precisely what Searle (1980, p. 37) requires, namely, "some independent reason for supposing that the rules are functioning causally": the apparent theoretical indispensability of appeals to grammatically characterized internal states in the explanation of linguistic behavior is surely the best sort of reason for attributing to these states a causal role in the production of behavior. Certainly a speaker's ability to state the rule that he took himself to be following would provide no better evidence. Even if we suppose that such an ability evidences an introspective awareness of one's mental states and processes, the methodological justification of explanations of such cases in terms of knowledge of a rule would rest on the prior assumption that appeal to rule-characterized internal states is theoretically indispensable. It is not the fact that we have explicit knowledge of rules that sanctions explanations in terms of rules; it is rather that we as postbehaviorists know that explanations of behavior must advert to internal processes and we know of no other way of characterizing these processes except in terms of mentally represented rules.

4. Computational reduction. Our construal of M leaves open the question of the computational realization of a speaker's knowledge of grammar. In particular, it leaves open the possibility that the mediation of linguistic behavior effected by such knowledge may be realized by something other than a representation-using system of the sort characterized in H3. Indeed, the realization might be extremely abstract, in the sense that there may be no computationally definable answer to the question "What specific computational structures or processes realize the speaker's grammatical knowledge?" In such an eventuality there would fail to be the sort of homomorphism envisioned by H2 between the mental processes imputed to the speaker by a grammatical characterization of his linguistic knowledge and the computational structures or processes attributed to the speaker on partially independent grounds. Such an eventuality would reflect the fact that the particular generalizations to be captured at a computational level of description must be stated in a vocabulary that cross-classifies grammatically characterized processes. Which of the possible realizations is the one instanced by human speakers is, of course, an empirical issue. Stabler correctly points out what sort of evidence would bear on the claim that human speakers realize a program-using system of the sort described in H3. What remains unclear is the distinction he proposes to draw between H3 and Hd: What is the distinction between a system whose computation of some program is governed by an encoding which includes the rules of some grammar and a system whose computation of the program uses the grammar *as data*? What sort of evidentiary considerations would distinguish these two sorts of system?

When do representations explain?

Daniel C. Dennett
Philosophy Department, Tufts University, Medford, Mass. 02155

Stabler's patient and insightful clarification of this issue is most welcome, for he has provided some quite visible and stationary landmarks against which to detect the ideological slippage that has so far marked this debate. He is reticent, however, in answering one dialectically important question: if, as he convincingly argues, none of the evidence to date supports H3, what sort of evidence conceivably *could* support it?

For a representation to figure as a representation in a causal explanation, it must occur in a context where it is "read" by some agent, organ, or device. What could establish that such a process occurs? Consider an obvious case: old Mother Hubbard lies dead on the floor, a victim of poisoning, an open and half-full bottle of paint remover in the cupboard. Acquaintances say she had not been depressed, but had complained recently of "fainting spells." "Aha!" says the detective, noting her thick eyeglasses. "The bottle label says 'FOR PEELING PAINT' and she must have misread it to say 'FOR FEELING FAINT.' See how like Fs those Ps are."

What could conceivably convince us that actual rule consultation occurs in language processing would be evidence that on occasion rules are misread. But evidence for this would require some extraordinarily hard-to-acquire sorts of supporting evidence: evidence about not just the function or operation of the rules (as Stabler shows), and not even about just the "abstract" form of the rules (for, as Stabler shows, this evidence is always reinterpretable as evidence about function), but about the actual physical features of the encoding and the reading mechanisms – not just the semantics and syntax of the language of thought, but its orthography and typography as well. Could anything less give us clear evidence in support of H3 over its more modest rivals? So far as I can see, nothing else would be *direct* evidence.

The trouble is that it is not clear, given Stabler's treatment of the program/data distinction, that even this sort of evidence would satisfy him. For how could we distinguish, given this incredibly strong (imagined) evidence, the alternative hypothesis that we had not simply uncovered the typography of the data-representing system, rather than the program-reading system? I am inclined to conclude that there is something fishy about Stabler's attempt to make that distinction, at least as it would have to be adjusted to be transported from computerland to psycholinguistics. Consider another simple case: we teach somebody a simple algorithm for performing some congitive task, such as deciding whether to open the bidding in bridge, or winning at Nim. This, then, will be a paradigm case of someone – in this case consciously, even self-consciously – consulting a remembered rule and guiding calculation by its lights. Is it a case in which the rule counts as data – "the rules as argument" – or as program? Perhaps the answer is obvious, but it was not obvious to me what Stabler's answer would be.

Supposing this point clarified somehow, we might return to the question of whether there might be *indirect* evidence strongly supporting the existence of represented rules. One possible line of argument, hinted at but skirted by Stabler, is one form or another of the "you can't get there from here" argument. One *might* argue, that is, that while "hybrid" and "hardwired" systems are always possible in principle and even, once created, faster and more efficient, they can only be created "naturally" by a design process that first implements a system in which the rules essential to the "rationale" of the system's function are explicit, and explicitly consulted. I think this is a risky and dubious sort of speculation, but its rationale is probably worth exploring. Consider the advanced bridge player who no longer consciously "counts points" (and who might not be counting them unconsciously either); there is surely some plausibility to the idea that the sophisticated but *ex hypothesi* merely H1 rule-described behavior of this player could only have been entrained by a process that includes an interim stage of H3 rule following. In a similar vein, one could argue that it is no accident that sophisticated hardwired microchips – such as those to be found in arcade video games – are designed by a process that begins with a program-guided system in which the operations are debugged. Tempting as these analogies are, however, they serve in the present context to highlight one of the most compelling sorts of indirect evidence *against* any H3-type theory of human linguistic competence. Surely the evolutionary design process that yields our innate linguistic competence as its product is strongly disanalogous to the design process that yields video games, precisely in being undirected, unforesighted, and completely lacking the sort of explicit "top-down" goal that is the hallmark of design (or training) processes that are aided by explicit "rules for beginners." [See also Dennett: "Intentional Systems in Cognitive Ethology" *BBS* 6(3) 1983.]

A few analogies with computing

Maurice Gross

Laboratoire d'Automatique Documentaire et Linguistique, University of Paris 7, 75221 Paris Cedex 05, France

The point of Stabler's whole discussion is not clear to me. Other linguists will presumably also wonder about the interest of arguing that just in case some informal remarks made by Chomsky are interpreted in a specific way, then they are wrong. Stabler perceives that, if these remarks are indeed correct, then linguists and psychologists are left in a vacuum about the place of a grammar in the brain; it is not clear, however, what type of research is affected by this possible vacuum.

There is no question about the correctness of Stabler's argument, but in the course of the discussion, certain undiscussed questions come to mind that overshadow and submerge the special argument he has made.

First, I do not see why linguists should currently be concerned about the way rules are represented, encoded, or generated in the brain. Linguists are still debating about the existence of particular rules and about their formal nature. These debates are concerned more with the empirical level of testing acceptabilities of sequences than with Stabler's speculations.

Specialists in computational linguistics are closer to the present debate. For example, the position that a grammar is a set of data used by a program is not as marginal as Stabler appears to think. It was one of the central ideas among specialists in automatic syntactic analysis in the 1960s, when the main application was mechanical translation. There have always been roughly two main schools of thought about the construction of the required algorithms: build fast algorithms, which implies directly using specific grammatical features of the particular language under analysis; build algorithms that are as language-independent as possible – such algorithms would not have to be modified when the grammar is extended or corrected and might apply to several languages, perhaps even to all.

These two positions clearly raise an important issue not often recognized by linguists: Where should the dividing line between the grammar and the recognition algorithm be drawn? The placement of the dividing line may change the form of the grammar substantially. Consider, for example, the case of context-free grammars, as discussed by Joshi, Levy & Takahashi (1975). These authors have shown that part of what seemed to belong to the essence of context-free rules and of some transformational ones could be moved to a recognition algorithm, the context-free rules themselves then becoming general conditions on sequences.

Another alternative that computer sicentists face daily also seems to be relevant to a discussion about the representation of grammars. Given a certain function, either its computation can be implemented through an algorithm, which is usually a compact program that is applied a large number of times, once for each computed value; or else the computations can be done beforehand, that is, the function can be put in the form of a (possibly) large table, and the algorithm that computes a given value is then reduced to a simple table–lookup procedure. In the first case, the memory occupied in the computer is small, whereas the processing time is high (equivalently, the speed of the computer has to be high). In the second case, the memory needed is large, but the processing time is shorter.

Such concrete questions are ultimately empirical, but they could also be discussed *in abstracto*. If one is eager to pursue an analogy between the brain and the computer, a number of similar proposals appear to be relevant.

The recent technology of very-large-scale integration of circuits might suggest new processes for storing and/or acquiring a grammar. Consider the way specialized chips are now being built. A program is written in a high-level language such as PASCAL. This program is then compiled and *automatically* transcribed into charts that will be photographically engraved on a chip. At this point, the differences between programmed and wired instructions become vanishingly small, perhaps to the point that Stabler's three levels may no longer be relevant concepts. The way such chips are made indicates that a grammar could be acquired just by reading it once, as in the thought experiment Stabler mentions.

One could also enter into the distinction between compiler and interpreter for high-level languages. A grammar may correspond to either type of device. There may be interesting analogies between the way compilers (and whole systems) are built and the acquisition of grammar. A compiler, for example, is built in several stages. First, a core containing a few essential functions of the language is defined and is written (represented) in machine language, then, the rest of the compiler is written (represented) in the high-level language itself. There is no longer any need to write this part of the compiler in machine language, since the core compiler can carry out the translation. A discussion of the acquisition of a grammar might be modelled after such a device. At an early stage of learning, a core grammar, which may be partially innate or acquired, would be actuated. Then, the bulk of the language would be acquired at a different level, as the result of processing by the core grammar.

Other computational devices can be used to evoke linguistic processing, but I still cannot see how linguists can derive any useful insight from discussions à la Stabler.

In fact, the analogy between brain and computer on which the target article is based has been considered extremely dubious by many investigators – at least those outside the field of artificial intelligence. The speeds and rates of transmission of signals in the two systems are not comparable, and it is by no means clear that the large amount of parallel computing that appears to go on in the brain can compensate for the differences of transmission time (e.g., Crick 1979). Moreover, it has often been noted (e.g., by Chomsky & Miller 1964) that computers are too versatile to constitute valuable models of linguistic behavior, hence of the brain. Computers are known to be rather ineffective, for example, in recognizing syntactic patterns, whereas they are extremely efficient in computing solutions of differential equations, a task the brain is not adapted for. In Stabler's terms, the brain does not have the plasticity computers have.

Internally represented grammars

Gilbert Harman
Department of Philosophy, Princeton University, Princeton, N.J. 08544

Stabler compares a "second-level" hypothesis about grammar with a corresponding "third-level" hypothesis. According to the second-level hypothesis, users of a language "compute" or (better) execute the rules of a generative grammar in generating sentences; first they decide on the category, sentence (S), then they decide the sentence will be a noun phrase (NP) followed by a verb phrase (VP), etc., finally ending up with a derivation of the relevant sentence. According to the third-level hypothesis, speakers have an internal representation of the rules of generative grammar which they use when they talk or listen to what others are saying. Stabler claims that a third-level hypothesis is stronger than the corresponding second-level hypothesis and that linguists like Chomsky write as if they hold the stronger third-level hypothesis while arguing only for the weaker second-level hypothesis.

These claims are mistaken in several respects. First, the relative strengths of the second- and third-level hypotheses are reversed. A second-level hypothesis is stronger, since it implies the corresponding third-level hypothesis. That this implication holds is perfectly obvious from Stabler's definitions, given the further trivial observation that any mechanism that computes a function is naturally treated as a representation of that function. In particular, if speakers actually did first decide on a sentence, then on NP followed by VP, and so on in accordance with the rules of a given generative grammar, that would be the strongest possible way in which the grammar might be internally represented. Since Stabler thinks there are reasons to think that such a second-order theory is correct, he is logically committed to thinking that grammar is internally represented, since that conclusion is entailed by what he concedes.

The problem – and this is the second thing wrong with Stabler's argument – is that Chomsky and his associates precisely do *not* argue that speakers generate sentences in this way by executing the rules of generative grammars. Chomsky has observed in many places that the term "generative" in the phrase "generative grammar" has no such implication. The relevant sort of generation is mathematical, not psychological. Chomsky's claim, then, is that grammar is internally represented whether or not speakers generate sentences in accordance with its rules. He is arguing for a third-level theory that is weaker than the corresponding second-level theory. Therefore, Stabler's entire framework is misguided and completely irrelevant.

Stabler's remarks about plasticity are similarly irrelevant. Throughout his article, with an occasional disclaimer, he suggests that plasticity is a sign of representation, in contrast with circuits that are "wired in." But consider a personal computer with its ROM (read-only memory – unmodifiable), RAM(random-access memory – modifiable), and on–off switch (modifiable). Data and programs, internal representations, reside both in ROM and in RAM; whether they are plastic or modifiable makes no difference to whether they are internal representations. Nor does the fact that one can turn the computer on or off mean that the setting of the switch represents anything!

In order to see what Chomsky is getting at one needs to look at the details of linguistic theory, something Stabler does not do, and something I cannot do in the space allotted me. But let me at least point to where to look. The key is that the hypothesis:

(H) Grammars are mentally represented

has played an important role in linguistic research by Chomsky and his associates over the last thirty years, research that has led to many important discoveries about language. It is difficult to see how these discoveries could have been made if H had not been accepted. So these discoveries provide at least some confirmation of the hypothesis H and perhaps give meaning to it.

More precisely, the relevant hypothesis is:

(HH) The mental representation of a language user's grammar is similar to the sort of representation a generative grammarian would devise for the language.

HH is an important presupposition of the most salient feature of Chomsky's work, namely, that from the study of a single language, English, Chomsky has been led to surprising hypotheses about universal grammar whose approximate truth has been confirmed through the later study of other languages. Chomsky has observed that, if HH is true and speakers of English internally represent principles of grammar that resemble those principles he and his associates have formulated, the speakers do not learn the principles in any sort of ordinary inductive way, since most speakers are never exposed to relevant evidence. But if the principles are not learned in this way, they are presumably there from the beginning. This means that would be part of whatever grammar was internally represented, whether of English or of any other language. The principles should therefore be true of all languages. So, given HH, the study of one language leads to surprising predictions about other languages, predictions that, as I say, have been found to be for the most part approximately correct. Since it is difficult to see how these predictions are forthcoming unless HH is assumed, this is strong evidence for HH.

I am oversimplifying, of course, since there are various ways in which a principle might be internally represented even though users of the language had no evidence for it. A principle might represent the "unmarked case" – it might be a "default" principle, one that language users start with and keep in their internal grammars unless they get positive evidence against it. And there are other possibilities that are being actively investigated in current work in linguistics (Chomsky 1981; 1982). But the basic idea is still the same. And the principle HH has a crucial role in this ongoing research.

Computational commitment and physical realization

Robert M. Harnish
Departments of Philosophy and Linguistics, University of Arizona, Tucson, Ariz. 85721

When one claims that mental states, events, or processes are "computational," what is one claiming? And when one claims that some human behavior is "rule-governed," what is one claiming? If being rule-governed entails having rules (or representations of rules) that are causally efficacious in the production of (some) behavior, then it is natural to treat these questions as related, because one of the better-understood systems that can be said to be both computational and rule-governed is an ordianry "Von Neumann," "stored-program," "register-architecture" computer. Whether or not such machines have (or can have) the appropriate functional architecture to strongly simulate interesting human cognitive capacities (see Pylyshyn 1980), we desperately need to understand better how *any* cognitive capacity can be realized in a physical system – how the same system can receive both a physical description and a "functional" description in such a way as to explain in physical terms what it is doing in functional (cognitive) terms.

Viewed this way,the present fundamental contribution of computer science to cognitive science is not via artificial intelligence and (basically "weak") simulation, but rather via systems design and electrical engineering. Although clever programs have been getting most of the play, computer science via its "hardware" and "firmware" side, not its "software" side, has given us clear and well-understood instances of functionally describable capacities being realized *without remainder* in physical systems.

Most programs now run directly only on "virtual" hardware, but the relation between a program and the next level down (the virtual hardware) is not like the relation between the lowest level of software and firmware (read-only memories or "ROMs," for instance) to hardware. The former is like translation; the latter is more like simultaneous description, in that no independently motivated feature need be preserved by the mapping between descriptions. In other words, it makes fairly precise sense to speak of "translation" between programming levels, but no sense when referring to machine code and electrical engineering.[1]

What, then, *is* the relation between "hardware" and "software," if not translation or interpretation? Stabler finds three grades of computational involvement for physical systems: (1) computing a function, (2) computing an algorithm for computing a function, and (3) using a program to compute an algorithm for computing a function. If cognitive capacities are like running programs, then what evidence is there for this third level of description? Stabler finds evidence only for the second level of description. Though he may have missed some crucial sequence of experiments, it would probably be impossible at the present time to state enough independently motivated constraints on permissible psychological mechanisms to guarantee that any set of experimental data would decide between a second-level and a third-level theory (see Pylyshyn 1980).

The central value of Stabler's excellent article is to formulate computational commitment precisely. This illuminates a wide range of psycholinguistic and linguistic theses. For instance, the idea that grammars are psychologically real has received an immense amount of exposure (see Chomsky 1980b), yet what reason have we ever been given for supposing that theoretical structures, principles, and rules formulated in accordance with accepted *linguistic* methodology (mainly intuitive simplicity over a data base of intuitive judgments) should be similar to structures, principles, and operations formulated in accordance with the accepted methodology of *cognitive psychology?* Of course, it would be nice if linguists and psychologists could take in each other's laundry, but Stabler's framework makes it quite clear how difficult it would be to show such a thing, and so why it should be such a tenuous article of faith.

Finally, it would be useful to have a fourth level of description: a "programmable system." Given that this is one capable of being a program-using system, if we had an account of this capacity we would have a way of characterizing the difference between a system that was capable of being programmed to do such and such, but was *not* so programmed, and one which directly computed an appropriate algorithm. Such a characterization might help in assessing recent nativist accounts of concept development (see Fodor 1981c).

NOTE

[1] For instance, consider the case of "field-programmable" ROMs, where the customer buying the system uses current pulses to *blow out* selected portions of an otherwise equipotential circuit, thereby yielding a physical system with the desired functional description. See F. Hill and G. Peterson (1981), ch. 8.5.

Levels of grammatic representation: A tempest in a teapot

Michael R. Lipton
Department of Philosophy and Religion, Northeastern University, Boston, Mass. 02115

It is a mark of some progress in the discussion of the role of grammar in the expression of linguistic abilities that certain questions are not asked. In particular, Stabler finds it meaningful to attribute representations of this or that piece of information to people and computers; he thinks we encode information and execute algorithms on these representations. Also, he recognizes that the question of the form these representations take and what algorithms run on them is an empirical question. This means that there is no a priori proof of the answer, whatever it happens to be. A particular answer is a good one if it is justified by what is politely called inference to the best explanation. I detect a bit of backsliding on this issue when he argues that any system explained at level three can be explained by a level-two theory. But this claim is irrelevant if the issue is empirical.

The chief difficulty I have with the paper is that Stabler, I think, seriously misrepresents the evidential situation. He argues that there is evidence for H2, and that the evidence now available does not distinguish H2 from H3, and therefore we have no evidence for H3, a conclusion he draws often. But this is a mistake. Positive evidence for one of two hypotheses which does not distinguish it from a second is surely evidence for that second. He is right that little is available to distinguish H3 from H2, and that is because so far little is thought to depend upon the difference.

The argument for the second-level hypothesis (H2) is that, given it, we can explain why the language satisfies the rules of grammar G, and how a person understands his language of which G is the grammar. Likewise, we might motivate a third-level hypothesis, in which G is actually encoded or represented in a person, a hypothesis that seeks to explain the acquisition of the cognitive capacities explained in the previous level-two theory. Thus, this hypothesis will not be H3, which only concerns comprehension. If to understand a language one must come to compute G, then to learn the language one must compute some other function the output of which is the state of being able to compute G. Regularities in the course of learning, or the state attained, or the dependence of the state attained on experience, all may best be explained by attributing to the learner various algorithms working on grammars. Essentially, this is the argument Stabler likes for a level-two theory of understanding made for a level-two or level-three (I can't tell which – see below) theory of acquisition, which has grammars represented.

The plausibility of this argument for a level-three hypothesis strongly depends upon the form that learning theory takes, but there surely are learning theories that approach matters in this way.

The plasticity of learning, such as it is, suggests a third-level hypothesis, although, as Stabler points out, it will not distinguish a case of G as part of the program from G as part of the data. In thinking about the case for actual computers, it seems that plasticity argues in favor of an encoded program because we know how the environment affects the programs that can be computed. This is exactly what we do not know in the case of language learning. If a system can acquire many different capacities, it might satisfy a level-three theory for each of those capacities, but it needn't. It is only under many assumptions about the nature of learning that a level-three theory is supported, assumptions about the fixedness of the hardware and the degree of innateness of the capacities. It seems that any argument from plasticity really depends upon the available learning theory, and not on considerations of how varied the acquired capacities are.

Someone interested in defending the mental-representation hypothesis might well have in mind a thesis like Hd, which specifies that the grammar of a language is used as data. This hypothesis appears to be different from H3; yet this difference is difficult to characterize and is not characterized by the distinction between level two and level three. Consider a standard machine running a program in a higher-level language through an interpreter. The higher-level program is a program represented in the machine and controlling the computation; yet since it is being run in interpreted form it counts as data for the interpreter. The point of the example is to show how unclear the intended contrast is, even in cases we think we understand, and that it is doubtful that at our current level of understanding the difference makes a difference.

Whether a grammar G is part of a level-two theory or a level-three theory has not been substantially addressed, to my knowledge. What theorists actually believe can best be gleaned by looking at concrete proposals for parsers, compilers, and learning devices. Even then it isn't clear that the level-two/level-three distinction is of much importance. It is hard to see that anything hinges on it, especially since we do not have very good ideas about the algorithms adequate to instantiate various linguistic capacities, irrespective of how they are implemented. It is doubtful that any long-range harm will come from assuming or theorizing as if a level-three theory were correct. If we ever got a level-three theory of a linguistic capacity that was materially adequate, we would know how to replace it by a level-two theory.

On speculating across opaque barriers

Abe Lockman

Department of Computer Science, Rutgers University, New Brunswick, N.J. 08903

Stabler's arguments raise a general question: what sorts of hard conclusions *can ever* (i.e., in principle) be drawn concerning the internal workings of mechanisms (whether animal-, vegetable-, or mineral-based) from mere observation of their external input/output behavior? Assume that we are observing some mechanism M (e.g., a natural language parser) whose insides are invisible to us: that is, we can only see pairs <1,0> of its input and corresponding output. In such a situation the only certain statement that we may make concerning M is an observation that its output O is some function F of its input I. If we observe that mechanism M computes, in this sense, some function F, then there is in principle no way to tell whether it does so via a nondecomposable input-to-output mapping (e.g., table–look-up), via a decomposable but immutable (for a given input) series of computational steps, or via the execution of a potentially changeable program; thus there is no hard basis for supporting a preference for any of what Stabler terms first-level, second-level, and third-level theories. Certain additional assumptions, however, lead to the following exception to this. If we believe the mechanism to possess limited storage capacity, and if the function that it computes is over an unbounded domain (as most competence theories of grammars maintain), then we may eliminate the possibility of a first-level theory, since decomposition into recursively applicable steps is a minimal requirement for computation over unbounded domains using only limited storage. Also, of course, the existing theory of formal languages and their corresponding (in computational power) automata allows us to distinguish between different *sorts* of second-level and third-level theories for different functions computed. (All of this is not, of course, intended to imply that particular theories of internal composition and states may not be extremely useful as a way of characterizing function F; although we cannot prove that atoms exist, for example, it is convenient to describe the world as behaving as if they did.)

The above observation subsumes Stabler's arugments against two possible supports for a third-level theory of language use: the explanation of basic linguistic abilities and the commitments of formal linguistic universals. The crux of the matter is, then, what may be inferred from the nature of language acquisition, where we are observing how an input-to-output function changes over time, rather than a static function. Here I must disagree with Stabler and claim that the phenomenon of language learning (or "plasticity") does, in fact, support a third-level (i.e., program-executing) theory of language use.

Now the entire point of constructing (or possessing) a programmable mechanism is that it can "learn" in the following sense: rather than computing a fixed function F, we can feed it any of a variety of programs P, enabling it to compute (having "learned") any of a variety of functions F_p of its inputs. Given that humans can, in fact, learn any human language, one can always view a human language user as a programmed system: language acquisition amounts to absorbing knowledge of a particular language, which is then used as a variable program by a general-purpose language-using mechanism. While Stabler agrees that support for a third-level theory should be sought in this aspect of language usage, he puts forth several objections to the above sort of argument, which I shall deal with in turn.

First, Stabler proposes that what appears to be learning by a mechanism may merely be the effect of very complex hardwired internals rather than of a programmed system. If this were the case, it would imply that complete linguistic capability for all languages exists in humans from birth, i.e., that nothing need be learned. This contradicts all observed behavior. Second, he suggests that language learning may consist of the "growth" of special submechanisms in the brain, or "only certain adjustments and additions" to an existing hardwired mechanism. Whether such growth or adjustment is viewed as actual neural growth/change or simply the encoding of information in existing neurons, it still does not change the fact that some initial portion of the brain will subsequently execute whatever is built or otherwise encoded. A program may be stored in many different physical ways, which can differ in the difficulty (or even possibility) of initial storage and/or alteration.

Finally, Stabler suggests that, while plasticity would seem to require some encoding E of knowledge of a language, one cannot distinguish whether E is itself a program or "whether it is used strictly as data by some other procedure." Now one of the basic notions of computation is that of program/data equivalence. That is, for any program P, written in some language L and computing some function F, one can also implement F by another program P′, written in some other language L′, which uses P as its data and as its sole knowledge of F. A distinction between program and data requires that one know the language which the mechanism under consideration implements; then the program is that portion of the mechanism's storage which consists of statements in this language which are to be executed, while the remainder is data. (Stabler himself makes no distinction between program and data in his initial definition of a third-level theory.) Thus, if one has no knowledge of the internals of the brain, one cannot usefully argue as to whether encoding E is, in fact, program or data. Any such claim would require knowledge of what the "machine language" of the brain is, i.e., what set of basic "instructions" it will recognize and execute. Such knowledge is, I believe, still a good distance in the future.

Execute criminals, not rules of grammar

James D. McCawley

Department of Linguistics, University of Chicago, Chicago, Ill. 60637

Linguists in the 1930s and 1940s talked about stimuli and responses and then proceeded to examine linguistic examples, identify linguistic units, and formulate rules describing how those units could be combined. Linguists in the 1960s and 1970s talked about computability and algorithms and then proceeded to examine linguistic examples, identify linguistic units, and formulate rules describing how those units could be combined. The professed behaviorism of the former linguists and the professed commitment of the latter linguists to computational models are not totally irrelevant to the kinds of linguistics that they did, but it has a much more tangential relation to their work than most of them would have liked to think. My impression is that the principal influence of these ideologies on the scholarly work done by their adherents was on those scholars' judgements as to what ideas posed "conceptual problems" (in the sense of Laudan 1977) and were thus to be rejected unless they could be rendered ideologically palatable.

I have argued (McCawley 1982) that linguists' statements as to the nature of their enterprise should be regarded with suspicion (I leave the drawing of similar conclusions about other disciplines to those familiar enough with the disciplines to have grounds for drawing them). For example, while most transformational grammarians assent to the proposition that a language can be identified with a set of sentences, the practice of most of them strongly suggests that they do not believe that proposition. Stabler takes his conception of "a grammar" exclusively from linguists' (indeed, very orthodox transformational grammarians') statements as to what a grammar is supposed to be, and he fails to take up what linguists actually accept as (partial) grammars of languages. In view of the predilection of transformational grammarians for computational terminology and for notational systems reminiscent of programming languages, it is not surprising that Stabler treats a grammar as something that can be included in a program and its rules as things that can be "executed." However, such a conception of a grammar conflicts with the practice of most linguists, even of the few that Stabler mentions, who admit grammars in which some or even all the rules fail to be the sort of thing that one can speak of as being executed.

For example, unless one is very loose either about what it is to execute something or about what one identifies with a given linguistic rule, one cannot speak of executing a "filter" such as Chomsky and Lasnik's (1977, p. 456) rule excluding sentences in which certain complementizers are followed by empty subjects. Conceivably Stabler intends a loose use of his terminology, so that computational steps that indirectly reflect the content of a rule (say, a subroutine that serves to avoid the assembly of structures that violate a given filter) are identified with that rule, but he has given the reader no clue as to whether he has any such thing in mind. As they stand, Stabler's H2, with its reference to execution of grammatical rules, and his H3, with its reference to computations governed by an encoding of a program that contains a grammar, are irrelevant to the status of the sorts of grammars seriously advanced by linguists, in the same way that talk of executing rules of poker or of chess would be irrelevant to the status as part of poker-playing or chess-playing competence of the rule that three of a kind beats two pair or the rule that bishops move diagonally. I use the rule that bishops move diagonally when I decide not to move my rook to a square that is diagonally opposite my opponent's bishop, but I doubt that Stabler would want to say that I "execute" the rule.

Nine-tenths of the way through his paper, Stabler finally gets to the topic of "the grammar as data" but shortly dismisses it as of little apparent interest. By all rights it is that topic to which nine-tenths of a paper entitled "How are grammars represented?" should have been devoted, and the idea of a grammar as part of a program that should have been taken up only to be dismissed. Users of a language have knowledge (of which the linguist's grammar is intended as an account) of the phonological, morphological, syntactic, and semantic structures that the language allows and of the permissible correspondences among those structures. One of the few things that most linguists agree on (Chomsky 1965, p. 9; Lakoff & Thompson 1975, p. 307) is that the same knowledge of a language is used in speaking it as in understanding it. The computations that one performs in speaking, in comprehending speech, and in reading, however, are presumably quite different from one another, at the very least because of major differences in what information one has access to (for instance, in speaking, but not in comprehending speech, one has direct access to the speaker's intentions), though in each case one is assembling linguistic structures on each of the structural levels, guided by the same knowledge of what structures are possible and how the different structures may be related to one another. The fact that older-style transformational grammars (as in, say, Chomsky 1965) are presented in a format reminiscent of computer programs should not delude one into ascribing to rules of grammar the status of computational steps to be executed in the course of linguistic behavior. The information that transformations provide about the correspondence between underlying and superficial linguistic structures can be utilized by the language user (speaker *or* hearer) before he finishes assembling the structure that the transformation "operates on," in which case the rule is used, though the "instruction" by which it is expressed is not executed. The notational practices of transformational grammarians can be compared to the imaginable practice of giving information (such as that there is a bridge connecting 59th Street with Long Island City) in the form of instructions ("Go east on 59th Street and over the bridge and get off in Long Island City") even when there is no intention that the instructions be complied with. (If a grammar is, as I am suggesting, more like a map than like a program, the arrows that appear in rules are like the arrow on a map: they tell you which way is north, with the convention that surface structure is north of deep structure.)

Stabler's dismissal of "the grammar as data" is merely a baroque elaboration of an argument from ignorance. Stabler states that "proponents of M are never so explicit about the role of grammar, so it is really more plausible to assume that they are proposing that either H3 or Hd is correct" (M = that grammars are mentally represented; H3 = that language understanding is done by a program-using system with a program that includes a grammar; Hd = that language understanding is done by a computational system that uses a grammar as data); this "more plausible" assumption, however, "seems rather unnatural" and, since it is "looser" than H3, it is also undermined by the arguments that undermine H3. Stabler's reliance on a "plausible assumption" about what "proponents of M" have in mind seems to be an admission that he has not asked any "proponents of M" what they have in mind. A scholar who has spent most of his academic life at institutions where proponents of M are not hard to find ought to be able to rely on more than a mere conjecture as to what such persons accept. In any event, the inexplicitness of proponents of M about the role of a grammar is no more grounds for dismissing Hd than it is for dismissing H3: Stabler has not cited any proposal for the role of a grammar in language use that is both explicit and plausible, so any of the "competing" proposals (its's a bit ludicrous to say "competing," since the competitors seem more committed to withdrawing from the race than to running in it) could be given the privileged position that Stabler accords to H3, and its competitors replaced by "looser" substitutes. The dereliction of duty of which Stabler rightly accuses linguists does not justify him in putting inanities into their mouths, much as I might regard that as a punishment which fits the crime.

How could you tell how grammars are represented?

John C. Marshall

Neuropsychology Unit, Neuroscience Group, The Radcliffe Infirmary, Oxford OX2 6HE, England

I *think* I agree with most of the arguments that Stabler advances, but I'd like to be reassured – or disabused.

To begin with, it is clearly true that the research program of generative linguistics (and the "realist" interpretation of specific linguistic hypotheses) implies no necessary commitment to an encoded grammar in Stabler's sense. As Chomsky (1978) points out, grammars specify "abstract conditions that unknown mechanisms must meet," an interpretation of linguistic inquiry that, as I have previously argued (Marshall 1980), corresponds with Marr's "top level [that] contains the theory of the computation" (Marr 1980). This construal of the status of grammars certainly leaves pretty much open such questions as: What are the algorithms and heuristics employed in parsing and production? How are these algorithms committed to particular mechanisms? and, How are the mechanisms instantiated in neurophysiological circuitry? There are, of course, good reasons for expressing parsing algorithms, for example, as computer programs. It is, as Stabler notes, somewhat easier to write the theory in a high-level programming language than to build it directly in the form of "networks of electronic circuits." But there is no reason why this undoubted fact should lead to reification of programs per se. Bratley, Dewar, and Thorne (1967) have described a computer program "to simulate the process by which humans recognize the syntactic structure of sentences." Nonetheless, they conclude that it is important "to distinguish those features of the model which are significant from those which are merely a consequence of the fact that the model is a computer program" (p. 973). There is no implication in Bratley et al.'s paper (or indeed in any other paper on syntax recognition I am familiar with) that the neuronal instantiation of parsing theories must run on a program-using architecture. Similarly, the fact that low-level visual processes may be implemented directly is no bar to Grimson's (1981) expressing the Marr and Poggio (1979) algorithm for human stereopsis as a computer program.

How, then, could we decide whether particular linguistic computations "involve the use of an encoding of the program computed"? Stabler's own thoughts on this question seem somewhat gloomy. He stresses the fact that "any program that can be computed by any system can be computed directly, *or* by a program-using system, *or* by some sort of 'hybrid' system," and he suggests that the distinctions are really differences "of detail at the level of the wiring." But is there really a *principled* distinction between "direct" computation and programmed computation, as opposed to the technological distinction between taking a soldering iron to the innards and reprogramming from the outside? If a theoretically significant issue is involved ("at the level of the wiring"), it is surprising that Stabler refuses even to speculate on how "neurophysiological data could be brought to bear on any such issue." I would have expected that if the *conceptual* difference between program-using architectures and "direct" computation is as clear as Stabler maintains, it should be possible to state *in principle* what kinds of anatomical or electrophysiological evidence would distinguish between them. (I lay aside the question of whether our current neurophysiological techniques are sufficiently powerful actually to obtain the requisite evidence.)

Elsewhere, Stabler stresses the importance of "control states" or "a mediating control mechanism" for diagnosing that a program-using system is operative, but I am not sure that I would recognize the relevant control mechanism if I saw one. Consider traditional examples of stimulus ambiguity: O is represented as "zero" in the sequence −2, −1, 0, +1, +2, but as [ou] in the sequence M, N, O, P, Q. Can one regard *digit* and *letter* as control states that determine the encoding of the stimulus (Jonides & Gleitman 1972)? If I came across the word *mare* in a French newspaper I would assign it the semantic interpretation of "pond"; if I came across *mare* in an English newspaper the semantic interpretation of "jument" would be rather more appropriate. Does the (putative) existence of an "input switch" in bilinguals (Macnamara & Kushnir 1971) count as a "mediating control mechanism?" If "context" determines the parsing of "They are flying planes" as either (They) (are flying) (planes) or (They) (are) (flying planes), does this imply that the maintenance of semantic coherence in a discourse is achieved *via* a control mechanism that "decides" which encoding is to be computed? With respect to lexical ambiguities in sentences current evidence seems to indicate that initially all readings are accessed; in a second processing state inappropriate readings are then suppressed (Tanenhaus, Leiman & Seidenberg 1979; Swinney 1981). In Stabler's terms, does this mean that the first stage is "hardwired" but the second stage is "executed by a program-using system" with control states? Subjects in lexical-decision experiments seem to be able to exercise options on which code (visual, orthographic, phonetic, or semantic) they rely on to perform the task; within limits, the construction of these codes is under strategic control that is determined by the subjects' perception of the character of the stimulus ensemble presented in the experiment (Hawkins, Reicher, Rogers & Peterson 1976; Shulman & Davison 1977; Carr, Davidson & Hawkins 1978). Do such results point in the direction of a program-using system?

If none of these cases counts as an example of "a control mechanism that determines what program is to be computed," what would?

Word processor or video game?

Robert May

Department of Linguistics, Barnard College, Columbia University, New York, N.Y. 10027

Is the manner in which a person embodies and ultimately employs a grammar more like the DEC-20 computer running the word-processing program with which I'm writing this, or more like the Pac-Man game down at the local video arcade? The DEC-20, on the one hand, is a "system which uses [a] program," in this case EMACS, "to govern its application." It is what Stabler calls a third-level computational system. The Pac-Man game, on the other hand, is a machine of a somewhat different sort. The game is "wired in" to its circuits; it contains "no explicit encoding of a program" and functions "without having control states to govern [its] application." It is what Stabler calls a "direct" system. Now, to restate the initial question in the terminology Stabler introduces, is a grammar, as a psychological theory, a third-level or a direct system; is it like a description of the control states of the word-processing program or of the game's logic circuits? Stabler does not provde us with an answer, but then it is not his intention to resolve the issue. Rather, his goal is to point out that the evidence purporting to support a third-level interpretation is, for the most part, also compatible with a direct interpretation. While grammars certainly describe a certain function (and hence qualify as second-level computational systems), linguistic theory itself is indeterminate between whether grammars are physically realized in the sense of software or hardware; whether we are dealing with the word processor or the latest wizardry from Atari.

I am inclined to agree with Stabler's point, such as it is. As Stabler points out, which view is correct is an empirical matter; indeed, he is at pains to clarify the types of arguments that have been tendered. I, for one, am willing to let the chips fall where they may on this one. After all, Stabler is not claiming that linguistic theories lack psychological interpretations; rather, he is examining what such interpretations consist of. He leaves the basic goals of empirical adequacy for linguistic theory undisturbed. This is not to say, of course, that this issue is an uninteresting one; it is by no means uninteresting, especially when considered from the perspective of a biology of language. What we might expect to find upon investigating it, none too surprisingly, is that the software/hardware distinction can't be taken too literally. The DEC-20 is an all-purpose computer; I can just as well play computer games on it as do my word processing. Indeed, it can run programs of exceptional sophistication and complexity, provided that someone programs it. But how does a person get programmed for a grammar? If anything is pretty certain about the acquisition of grammar at this point, it is that its central principles are not "learned" in any sense, because the experience necessary to induce them is simply not available to the child in any systematic fashion. On the other hand, however, I can play only Pac-Man on that machine down at the arcade; there is no way I'll get to play Asteroids or Space Invaders for my quarter, short of altering its circuitry. People, though, speak literally thousands of tongues but presumably do not substantially differ in those aspects of brain structure and function relevant to language. Reality, then, seems to fall somewhere between these extremes; grammars are more like what Stabler calls "hybrid" systems. Insofar as a linguistic theory attributes grammatical universals ultimately to the genetic endowment of the organism, it is committed to at least certain aspects of the grammar being hardwired, not programmed, for how could a child come to know these principles, given the "poverty of the stimulus," unless they were intrinsic properties of the system? All that would remain open, on this view, would be how those properties which individuate grammars, the "software applications" permitted by the nature of the hardware, are instantiated. So it seems that the relation of higher-order cognitive theories such as grammars to their physical realizations is most like an Atari home video system; I can't word-process on it, but I can play Pac-Man, Asteroids, Space Invaders, and a host of other games just by putting in the proper cartridges.

But perhaps this is enough of this computer mishmash. Today's computers may prove no more helpful as metaphors for mind than did yesterday's hydraulics.

ACKNOWLEDGMENT
Thanks to Virginia Mann for helpful discussion.

On levels

John Morton
MRC Cognitive Development Unit, London WC1H OAH, England

Stabler has drawn attention, in some detail, to the advantages of keeping separate the different levels at which it is possible to give an account of a set of phenomena. In particular, he has pointed out that a true linguistic theory will not specify the human language-processing algorithm. In general, the message is that higher-level specifications never entail particular lower-level specifications. As Stabler points out, Marr and Poggio (1976) have already made this claim and suggested that the first-level hypothesis should be stated fully before second-level hypotheses can be developed. Stabler thinks "this itinerary is overly rigid," but he does not pursue this demur. Of course, in practice, the majority of psychologists operate at the second level. This can be because their understanding of the phenomenon they are trying to account for is incomplete or does not exist in a suitable form. Thus, psychologists who attempted to rely on the current "true" grammar, viewing it as their function for suggesting the processing algorithm for that grammar, would spend more time trying to follow the linguistic debates than getting on with their own thing. The escape clause, in practice, is based on the modularity hypothesis. If we believe that the human language-processing algorithm can be split into component modules, then the properties of some of these modules can be explored in the absence of a complete description at the first level.

This description of current practice seems to violate one of Stabler's other claims, that "a second-level theory . . . is necessarily a first-level theory." In fact, there is no violation, since it seems clear that, in the quotation, Stabler is referring to a *complete* second-level theory. A second-level theory that refers to only a subset of the domain covered by a putative first-level theory, such as grammar, would say nothing about the grammar, of course. If a second-level theory were complete with respect to grammar then the mode of operation of the former would be described in terms of the latter. There would, however, seem to be no requirement that the terms in the grammar bear any direct relation to the nature of the modules in the second-level theory.

The same arguments apply to the relationship between second- and third-level theories and to that between third-level theories and physiology. Stabler takes an intermediate position on the latter, claiming that "the computational account of cognitive processes would need to be very well developed before neurophysiological data could be brought to bear" on third-level issues. Mehler, Morton, and Jusczyk (submitted for publication) have argued a little more strongly along the same lines – in effect, on the assumption that the neurophysiological level can be regarded as a fourth level. The extent to which the assumption is valid remains to be discovered.

The relevance of the machine metaphor

Thomas Roeper
Department of Linguistics, University of Massachusetts, Amherst, Mass. 01003

Stabler's work seems to me to be not a criticism of, but rather a fairly lucid exposition of the cautious assumptions that linguists make (Chomsky, in particular) about the mind/body relation. Linguists are sure that some relation – but who knows what – must exist between grammars and processing, acquisition, or neurological systems. Stabler points out that if one follows a strictly constructed computer model, then it may not be correct to say that the grammar is "represented" or "encoded" or that "rule-governed" behavior is involved. Nor, one might add, does he show that grammars cannot be neurologically represented.

What Stabler is doing has been done before. It belongs to a school of criticism with a formula. Take a science that uses a mathematical notation, observe that it has a few open definitions, and then show that its realization in another domain (usually physical) has infinitely many logical instantiations, due to a few vague definitions.

The problem with this approach flows from an excessive affection for the computer metaphor. The crucial initial assumption is that linguistic theory is a deductive, axiomatic theory that works like a machine. In reality, the deductive model is a goal and not a current reality. Therefore current concepts simply fail to have the rigor needed to be subjected to computerlike logical extrapolations.

This is as it should be, because linguistics is evolving in just the manner in which every other science evolves. Generative grammar is comprised of partially systematic and partially intuitive notions, many of which are deliberately left open (like *subject*) and others which are genuinely mysterious (like *referential*, or *thematic*). I think the true goal of linguistic theory is to achieve a *natural* fit between mathematically conceived grammars and neurological models. Although every linguist remains in principle open to the possibility of radical differences between an atemporal, aphysical representation and a temporal, physical representation, the underlying belief is that there will be a natural and perspicuous alliance. No one knows what "natural" means, but no one is upset by baroque logical possibilities, since they are merely reflections of (we hope) small conceptual defects.

The spirit of the enterprise in reality is quite alien to the computer metaphor. Chomsky has occasionally remarked that eventually linguistic theory will simply be regarded as neurological theory, just as mathematical versions of physics came to be regarded as physics. If this approach is viable, then what the linguist should do is head along a group of promising concepts in the direction of physical representations. The concepts represent insights, left underdefined, which can connect to biological concepts as they evolve.

This is just what has happened. The "modular" view of linguistics has emerged. It is coupled explicitly with the least well-defined version of generative grammar (namely, government binding theory), which, however, seems "natural" in light of what we know about biology and is beautifully detailed and nuanced with respect to linguistic data. If this is the true goal, then it is quite satisfactory at present to say that some relation between grammar and a physical representation will emerge.

Stabler's critique really belongs to an earlier era when the machine model did dominate linguistics. He alludes to this implicitly by his own discussion of "plasticity" and the "modular" views of language acquisition which have emerged more recently. What does a current assessment look like? Syntax has in many respects been cast in a theory of filters or output constraints that would map naturally onto a static model. In this respect a dynamic "rule-governed" conception perhaps becomes unnecessary. However, other aspects of modular linguistics could, in my estimation, easily resurrect dynamic characteristics with a straightforward mapping onto the processing or acquisition models. In local domains, then, the mechanistic model may work. This leaves open the most interesting question: What is the right imagery for overall mental ability? It is simply too early to tell.

What I would like to see as a contribution of philosophers to linguistics is a sympathetic clarification of *central* concepts, not criticisms of speculative remarks about undeveloped parts of the theory. What kind of concept is *empty category, pronominal anaphor,* or *thematic relation*? What is the status of the "extensional" theory of language learning? Are there interesting sympathies between these notions and modular concepts emerging in biology? (See Lewontin 1983, for an interesting critique of the computer metaphor in biology.)

Stabler's discussion of an alternative "parallel-processing" model moves in the direction of bringing intuitions about neurology together with intuitions from linguistics. A richly detailed discussion of these matters would be a real contribution.

Grammars-as-programs versus grammars-as-data

Jerry Samet
Philosophy Department, Wellesley College, Wellesley, Mass. 02181

I agree with almost everything Stabler has to say in his target article. His distinctions between grammars as encoded, computed, or embedded in hybrid systems provides a sharper focus for questions about the psychological reality of linguistic and cognitivist theoretical constructs. There are a couple of points, however, where Stabler and I diverge.

Grammars-as-programs versus grammars-as-data. Although it's clear that grammars are psychologically important in both program and data systems, Stabler downplays the second as an interpretation of linguists' claims for the psychological reality of grammars. At one point he says that the data version of the role of grammars "is not so natural a construal of the linguists' suggestions, and it is not supported by available data, either." I agree that the empirical support is not there. Still, I'm not sure why grammars-as-programs is a more plausible reading. What sort of program would realize a grammar? Suggestions in the literature don't strike me as all that convincing. Certainly the rewrite rules of any natural language-processing program will look quite different from the rewrite rules of any familiar transformational grammar. There seems to be no important connection between the production of sentences by a grammar and the production of sentences by a speaker, and the analysis-by-synthesis models Stabler mentions have very little plausibility as components of comprehension systems, either.

One possible version of grammars-as-data that Stabler does not mention would be a system in which the grammar is acquired on the basis of universal grammar, and then serves as a data base for the development of a set of parsing and generating heuristics (which may themselves be realized in any of the alternative manners Stabler discusses). Once such a parser and generator are in place the grammar may be "expunged" or kept as a data source for a fail-safe mechanism of some sort. In such a system, the grammar would be represented as data but would not be used in ordinary linguistic functioning. Of course, if we speculatively accept grammars as playing this sort of role, it doesn't *follow* that they must be represented in any strong sense (not even as data); perhaps a temporary hardwired or hybrid system is what gets converted into a heuristic parser. Nevertheless, the possibility remains that we can work on discovering a program/function that maps a representation of a grammar into a (skeletal) parser. In the course of such research we may confirm that we *are* dealing with a function that maps a *represented* data base into something. Although I remain skeptical about the correctness of all these models that incorporate grammars into processing systems, they are empirical possibilities nonetheless.

A similar issue about grammars-as-data comes up in Stabler's discussion of rule-governed behavior. If I understand him, he is suggesting that if a system incorporates a grammar simply as data, then the system's behavior is not really rule-governed. He implies that if we *are* willing to call such a system "rule-governed," then "we could show, e.g., the behavior of the Sun–Mars orbital system to be rule-governed." I don't quite understand the argument he gives for this. Consider someone who learns to play Monopoly by studying the rules. Suppose he's involved in a game and he has to rehearse (to himself or aloud) the rules for mortgaging property while he carries out the transaction. It seems plausible to say that his behavior here is "rule-governed." In fact, those who worry about the correct application of this term of art usually take this sort of case to be a paradigm of rule-governed behavior. But here it also seems most plausible to say that the rules are stored *as data* and that there is a more general program of some sort that mediates their application in particular cases. There seems to me to be no reason why a cognitive system that uses rules that it stores only as data should not be considered rule-governed in the appropriate sense.

The burden of proof versus the burden of research. There is one more point that I think needs to be clarified. Stabler often argues like this: a proposed hypothesis H′ fits the data, but H″ and H‴ are at least as good; why then should we accept H′? In response to this line of attack Chomsky has suggested that an alternative to H′ be articulated and compared to H′. Stabler is right in noting that you haven't established H′ conclusively unless you've ruled out the equally plausible contenders H″ and H‴, but this is not really to the point. Chomsky's response might be best read as a suggestion about empirical methodology. The idea is that it doesn't really get us very far to challenge a proposed theory with the logical point that an alternative is constructible. It would be more constructive to actually begin to articulate the alternative theory and then see whether it gets any independent empirical confirmation or disconfirmation. The fact that for every encoded theory there exists a set of "merely computed" counterparts is unchallengeable, but it doesn't go very far.

Computation misrepresented: The procedural/declarative controversy exhumed

Henry Thompson
Department of Artificial Intelligence and Programme in Cognitive Science, School of Epistemics, University of Edinburgh, Edinburgh EH8 9NW Scotland

Although I am in basic agreement with Stabler's subtext – that the role played in human linguistic behaviour by rules of a grammar is unclear from a philosophical perspective, and at best underdetermined by the available empirical evidence – I cannot agree with the main thrust of his supporting argument. It is based on a faulty model of computation and depends on a distinction without a difference – between execution and use. The question he claims to address is whether or not mentally represented grammars play a causal role in linguistic behaviour. The question he *actually* addresses is whether that role is transparent or opaque. There are three points I wish to consider: the computational framework: the hypotheses H3 and Hb; plasticity and Occam.

1. The computational framework. Stabler's characterisation of a third-level or "program-using" system is at best confusing and at worst incorrect. He uses the phrases *state of the system* and *control state* without making clear their relationship, and in ways which violate their ordinary usages. As nearly as I can make out, Stabler uses the phrase *control state* to denote a function from some unspecified subset of the *state of the machine* to functions from (encodings of) program statements to *states of the machine*. A more traditional and much simpler use of these words would say that a program-using system implements a single interpretation function which maps from machine states to machine states, and that within machine states it is sometimes useful to distinguish control state from memory state. In the first instance this story is usually told at the machine-language level – it is easy to see, for instance, how it applies to Turing machines or simple microprocessors – but it is straightforward to extend it to the level of virtual machines interpreting higher-level languages. At the machine level, one can identify *instructions*, which are the principal determinant of overall system behaviour, while at a higher level of abstraction one might speak of *program statements*, as Stabler does.

The problem with Stabler's approach is that it encourages him to make a distinction between grammars as programs and grammars as data which just isn't there. One would have hoped that the advent of PROLOG would have dispelled this aspect of the procedural/declarative confusion forever, but apparently it hasn't. As Pereira and Warren (1980) make clear, a minor augmentation of the PROLOG virtual machine allows it to use both ordinary and decorated context-free phrase structure rules as program statements, thereby effecting a parsing program.

2. The hypotheses H3 and Hb. This leads us to the next point, which is whether Stabler's hypotheses represent resonable straw men or not. Although he presents H3 as a fair expression of the computational implications of M, it seems to me that with respect to the distinction he is trying to draw between execution and use it is *not* fair. The crucial question is whether a representation of the grammar functions causally in language processing – the nature of the virtual machine with respect to which that causation has effect is not in the first instance at issue. To the extent that linguists have thought about this, I believe that their bias has always been in the other direction from H3, rather towards Hd. Starting with Chomsky's involvement with the Harvard Syntactic Analyser (Kuno & Oettinger 1963), through cognitive grammar (Lakoff & Thompson 1975), lexical function grammar (Bresnan & Kaplan, 1982a), and generalized phrase structure grammar (Gazdar 1982), the computational linguistic models on which linguists have drawn have all been of the "parsing program plus grammatical data base" (read "grammar-interpreting virtual machines") sort, and to my knowledge no linguist has ever suggested that the grammar rules are themselves the program, as Stabler would have them do. Certainly no linguist has ever entertained Hb; to do so he would have to have adopted the sophisticated confusion of Stabler's analysis of computation.

3. Plactiicity and Occam. Stabler's attempts to counter the plasticity argument are particularly undermined by his confusion about computation. He attempts to equate the possible role of "grammar as data" in the linguistic system with the role of, e.g., "numbers as data" in a calculator. But the crucial point he has missed here is that in a calculator it is behaviour with respect to numbers which makes us recognise it as such, whereas in a linguistic system it is behaviour with respect not to grammars, but rather, e.g., utterances which makes us recognise it as such. Numbers play a causal role in the behaviour of calculators, and (*ex hypothesi*) grammars play a causal role in the behaviour of linguistic systems, but of an obviously different sort. M claims that grammars are a constitutive part of any system whose behaviour with respect to language leads us to judge it a speaker/hearer. No such claim could possibly be made about numbers and calculators. The plasticity argument and the argument from the multiplicity of linguistic functions are indeed only arguments based on Occam (parsimony) – such facts as we believe bear on these issues are most economically characterised in terms of a decomposition into system(s) and grammar. Without doubt this gives grammar a causal role in linguistic behaviour, which is all that M has ever been taken to mean, until Stabler tried to interpret it in the inadequate and inappropriate terms of his so-called computational framework.

4. Conclusion. The real problem with Stabler's position is that it has no force. Even if his computational analysis is valid, all it leads one to conclude is that there is no solid evidence for a causal role for grammar in human linguistic behaviour. But neither is there any evidence against its playing a causal role. As between these two positions – that some cognitive structure homomorphic to some formal grammar does or does not play a causal role in human linguistic behaviour – Stabler has contributed nothing except his scepticism, which I share. His computational framework has nothing to say about this issue – even if it were formally adequate it could not. All his efforts in that direction amount to the beginnings of an attack on a much harder, perhaps necessarily unanswerable question, namely, "What is the nature of the above mentioned homomorphism," or, "What is the virtual machine which interprets the cognitive structures which are its 'output' "?

Rules are not processes

Robert Wilensky
Division of Computer Science, Department of Electrical Engineering 'and Computer Sciences, University of California, Berkeley, Calif. 94720

Stabler raises an important methodological point. Given that the behavior of a system conforms to a rule, when are we justified in saying that the behavior is caused by a representation of that rule? As Stabler indicates, this is not a new question. Critics of transformational grammar's psychological relevance have been raising essentially this objection for years, namely, that there is a long way to travel between characterizing a behavior and understanding the process that gives rise to that behavior.

However, Stabler's argument is both wrong and unfortunate. It is wrong because of a misconception that one would have thought long banished by now. Stabler confuses grammars with processes, despite his protestations to the contrary. For example, Stabler's formulation of his second-level hypothesis states that "language understanding involves the computation of some program P, a program that includes the rules of the grammar, G, *which are executed to generate linguistic representations*" (emphasis added).

This statement suffers from a bad case of category error: one may conform to a rule or violate a rule, but it is meaningless to talk about executing rules. This is especially true of transformational-grammar rules, which are explicitly acknowledged as nonprocedural; as Chomsky has explained ad nauseam, his rules characterize rather than perform. My objection is not a semantic quibble. Rather, I am raising the issue of how it is possible for a rule to play a role in the operation of a program, if rules themselves are not objects that can be executed.

One interpretation (which is what I suppose Stabler intends) is that a rule may be reflected rather transparently in the structure of a program. For example, one way to construct a parser based on transformational grammar is to construct a program segment corresponding precisely to the application of each particular rule. (Some parsers for programming languages, notably those termed *recursive descent* parsers, work this way, e.g. Aho & Ullman 1972.) While it is tempting to do so, one should not confuse this segment of the program with a representation of a rule – the mistake that I believe Stabler makes. For example, the rule itself applies equally well to understanding and to production, whereas this piece of program is useful only in understanding; the program segment of necessity contains flow of control information, whereas the notion of flow of control is alien to true generative rules, and so on.

Rather than being a representation of a rule, this segment of program is merely *designed in accordance* with the rule. Of course, a hardwired device might also be designed in accordance with this rule, and might behave in precisely the same way as the program segment with respect to the task at hand. As Stabler asserts, such a hardwired device need neither access nor manipulate an encoding of a program. But in neither the hardwired device nor the program segment would a *rule* be accessed or manipulated. Thus a rule of transformational grammar would not be represented and causally efficacious in one case more than in the other.

One might object that, if the code that denotes this program appears to be isomorphic to a transformational-grammar rule, then surely the rule must be represented and must be playing a causal role. The problem with this objection is that, for it to be valid, the rule must stop being a piece of program and start being a piece of data. That is, all control information, etc., which would distinguish this and other rule-motivated program segments from data objects must be drained from them, and distilled into the underlying machinery that executes the program. But now the program is no longer being executed; it is being interpreted.

So we can have actual rules in a program only if the rules appear as data to an underlying interpreter. The problem here, insofar as Stabler is concerned, is that he explicitly rejects "rules-as-data" as a sufficient condition for a system's being rule-governed.

This point is perhaps the most startling of Stabler's contentions. The "rules-as-data" view is a straightforward interpretation of how transformational grammar rules might be represented and causally efficacious. In fact, it would seem to violate common sense to say that such a system was not rule-governed. Rejecting this would also cause one to reject the idea that my behavior was rule-governed if I participated in a game by actually consulting a written list of the rules of the game while playing it.

Fortunately, Stabler's argument here is faulty as well. His objection to this position is that interpreting such systems as rule-governed would enable one to interpret the behavior of a planet in orbit as being rule-governed by showing some interpretation under which this system encodes rules which are taken as arguments in the computation of some function. Stabler's own remarks in the beginning of his paper show the way out: computational description in general "is 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 that because of their design are conveniently described in computational terms and useful for this reason." Thus one cannot reject "rules-as-data" as being causally efficacious because under some bizarre interpretation any system can be a rule. The question is, Under what circumstances is it "clear and particularly useful" to regard them as such?

Implicit in the preceding discussion is the notion that programs and data differ more in degree than in kind. That is, every nonhardwired program is some interpreter's data. So rejecting "rules-as-data" systems as being rule-governed while accepting "rules-as-programs" systems as rule-governed is logically inconsistent.

But Stabler's remarks are even more annoying than they are wrong. He purports to be raising an objection to the general claim that "human behavior is 'rule-governed' in something like the way that the behavior of a programmed computer is 'rule-governed.'" One should not need to point out that the lack of evidence for Chomsky's claim that transformational-grammar rules are mentally represented can hardly be considered a refutation of the idea that human behavior is rule-governed.

The real issue, at least to me, is the relationship between the components of one's theory and the components of the mind. For example, the mind might directly reflect one's theory (as in the "rules-as-data" condition); it may contain components designed in accordance with it; or the theory may simply describe the behavior of a system (such as the mind) without otherwise having any direct bearing on the system's design. These distinctions, which bear on what I have elsewhere termed the *procedural adequacy* of a theory (Schank 1977), seem more cognitively relevant to me than the distinctions Stabler makes. After all, one might object to transformational grammar on the grounds that its rules may end up bearing no more relation to how people understand language than do Peano's axioms to how people balance their checkbooks. But one suspects that few transformationalists would be upset if their rules ended up "only" being explicitly represented as data to various mental programs, or "only" directly manifest in the actual hardware of the brain.

Author's Response

Computational theories and mental representation

Edward P. Stabler, Jr.
Centre for Cognitive Science, SSC, University of Western Ontario, London, Ont., Canada N6A 5C2

What Chomsky really means. A number of commentators claim that I have misinterpreted Chomsky. **Harman** argues that my entire discussion is misguided because Chomsky has pointed out that we do not need to think of the rules of a generative grammar as generating anything in the speaker; they simply characterize the language. Actually this is perfectly compatible with both my construal of Chomsky and the arguments I develop. I quote and agree with Chomsky's remark that "there are many possible ways in which . . . a program might make use of a stored grammar; it is a central problem of psycholinguistics to explore these possibilities" (1969, p. 156). The idea of H2 and H3 (see Table 1 of Response) is, perhaps, suggested by passages quoted in the text and remarks like the following: "To know a language, I am assuming, . . . is to have a certain mental structure consisting of a system of rules and principles which generate and relate mental representations of various types" (Chomsky, 1980b, p. 5), but it is, as I say in the paper, only a "crude first guess" about how the rules might be used. This view is something Chomsky agrees to. (**McCawley**, who wonders why I did not just ask the linguists what they mean, will be interested to know that I did in fact confirm this view in discussions with Chomsky.)

The hypotheses H2 and H3, although admittedly crude, serve for the purposes of my paper because nothing in my argument depends on any of the details of how a program uses the rules and principles of the grammar. I point this out after quoting Chomsky's remark, and I emphasize the point again in note 4. Adding detail to H2 and H3 to make them more plausible is unnecessary for my argument. The reader may add any detail he wants to H2 or H3 or Hb to allow for "filters" (as **McCawley** suggests) or for some nongenerative use of generative rules, and my argument will go through in the same way.

Wilensky urges that it does not make sense to think of a rule as part of a program. One would have thought that executing a simple rewrite rule like "S → NP VP" might involve transforming a representation of "S" into a representation of the sequence "NP VP." Wilensky's objection is that a line of a program contains information about flow of control that is not specified by the rule, but he does not give us any reason to reject the plausible position that this information could simply be added. One rule could simply be a line in a program in which one line is executed after the next; or it could be a line in a program which would, after executing this rule, execute some rule which rewrites "NP" or "VP"; or it could involve some more complicated use.

Table 1 (Response). *Hypotheses referred to in target article*

(M) The grammar is mentally represented and used in the exercise of linguistic abilities such as understanding speech and making grammaticality judgments.

(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 linguistic representations.

(H3) In human language understanding, the computation of P is carried out by a program-using system whose operation is governed by an encoding of P (and hence also of G).

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

(Hd) In human language understanding, a program P is computed (either by a program-using system or by some other kind of system) that uses G *as data*.

(Hb) In human language understanding, the program computed either includes G and is computed by a program-using system *or* uses G as data *or* both.

The computational model. A number of commentators also worried about the computational framework described in my paper. The predicates "computes a function," "computes a program," and "uses a program" are given fairly precise definitions. These definitions are not offered as an analysis of the meaning of these predicates as they are used in ordinary language, though I do think that there is significant correspondence between my defined terms and the ordinary ones. However, in my terms, every physical system computes a function. This is a literal truth, not just a metaphor or analogy. Consequently, to say that some particular physical system computes a function (under some f_R and in some circumstances C) is not to make a substantial empirical claim. However, the claim that the terms of my computational account are appropriate for certain psychological or neurophysiological theories is certainly substantial. And as I point out in note 1, computational theories make substantial empirical claims about physical systems by putting constraints on the realization functions, f_R and f_P, and by providing some indication of the relevant circumstances C.

Roeper says that the machine model does not dominate linguistics any more; now linguists look for connections with biology. References to computation are still prominent, though, in the work of some linguists (e.g., Chomsky 1980b; Gazdar 1982; Kaplan & Bresnan, in press b), so the issue is not dead by any means. And in spite of the rather striking but still very crude results on localization of linguistic function in the brain, there are good reasons for doubting that any close connections between linguistic theory and biology will be forthcoming. Indeed, it is a crucial advantage of the computational approach that it has a functionalist vocabulary that does not require a type reduction to physicalistic concepts (cf. Putnam 1967; Fodor 1981a, ch. 5), since this seems to be required in linguistics and in psychological accounts of language processing and visual perception (cf. Liberman et al. 1967; Ullman 1980; Keyser & Pinker 1980).

Gross suggests that the fact that the brain seems to work at speeds much slower than are found in digital computers indicates that the computer analogy is inappropriate. The computational approach sketched here, however, has no bias towards fast processing. Slow, parallel processes are handled as easily as the processing that happens to be very fast in our electronic computers.

Computing a program versus using a program. This distinction plays an important role in my argument, and a number of commentators were unable to make sense of it. **Harman**, for example, argues that I am mistaken in thinking that a third-level theory (according to which a physical system *uses* a program under some f_R and f_P) is stronger than the corresponding second-level hypothesis (according to which that physical system *computes* P under that f_R). He says that given my definitions and the "trivial observation" that any mechanism which computes a function is naturally treated as a representation of that function, a second-level hypothesis will always entail the corresponding third-level hypothesis. In this criticism, Harman not only is mistaken about the logic but shows that he has missed the main point of my arguments. I am not worried about claims that certain functions are represented (whatever that might amount to) but rather about claims that certain rules, certain syntactic objects, are represented.

As for the logical point, suppose that under f_R and in circumstances C, physical system S computes a program P with two instructions Ii and Ij. Can we deduce that under that f_R and some f_P, S *uses* P? We cannot, because according to my definitions f_P must associate the two instructions of P with two states of S such that S computes F_{Ii} because it is in f_P (Ii), and then it goes into a control state which makes f_P (Ij) govern the computation, and as a result of being in this state it computes F_{Ij} (which might be different from F_{Ii}). The existence of states that play this causal role in the computation is obviously not entailed by the second-level account. It could be that if we allow ourselves to define any program-realization function at all, we will (as a matter of fact, not as a necessary consequence of the second-level account) always be able to find states that will serve in the required role, but again the point I make in note 1 is important: the scientist will be interested only in certain realization functions. Without implicit or explicit contraints on these functions, a computational theory has no empirical content.

With regard to the use I make of the distinction between program computation and program use, and of the corresponding distinction between encodings used as data and encodings used as programs, there is an important point that I did not emphasize in my paper, but that should have been clear given the form of my argument. Remember that I argue against both H3, which says that grammars are used as programs, *and* Hd, which says that grammars are used as data. Thus it is no objection to my argument to point out that, in some cases, a particular encoding can be treated either as data or as a program. If the rules of grammar are not represented as data or as a program, then, as I point out, it is hard to imagine *any* computational construal of M. If there are any cases in which rules are not used as a program but are nevertheless used as data, then we can propose Hd. As **Samet** and **Wilensky** point out, in such cases we could still say that the system was rule-governed. Wilensky is right that such a claim is significant only given constraints on the realization functions. The only problem with this as a construal of M, then, is the relatively minor point that this would be a case in which the rules were not computed. The serious problem is that neither H3 nor Hb is supported by available evidence.

Thompson suggests that the program/data distinction cannot be maintained in a programming language like PROLOG (similar worries have been raised about more conventional programming languages; see, for example, Burks 1963, p. 105, on branch commands). The point in these cases is not that there is no way to distinguish program from data but that it is sometimes hard to decide how to classify a particular encoding; it can sometimes be classified either way. As I have noted, this point does not undermine my argument. In a "procedural interpretation" of PROLOG, it is natural to think of a set of clauses as a program, and the queries (unless they are 0-place predicates) provide the input to these programs in their argument places (see Van Emden & Kowalski 1976; Clark & McCabe 1979). Thus, if the clauses are rewrite rules (as in Pereira & Warren 1980), then the rewrite rules are part of the program being computed. The input to such programs is typically the string to be parsed. The virtue of PROLOG is that its procedural interpretation corresponds to a standard logical interpretation under which the clauses of the program are interpreted as premises (and can, accordingly, be thought of not as a program but as the "data base") and program computation can be seen as theorem proving (see, e.g., Stabler 1982; Stabler & Elcock 1983).

Harnish notes quite correctly that a program-using system need not be programmable. The existence of an encoding of a program and a control mechanism that uses that encoding to govern the operation of a system certainly does not imply that the encoding is easily changed. Harnish may also be right that programmable systems should be singled out for particular attention. They may play an important role in the sorts of adjustments in cognitive functioning that representation-using systems allow.

Computing compiled and interpreted programs. The fact that we sometimes compile a program and then run the compiled program does not raise any problems for the framework proposed, as **Berwick** suggests. If we use the original program to produce the compiled program and then load and run the compiled program, the present framework allows us to say precisely what has gone on. The original program was taken as data by the compiler to generate another program as output, and the program so produced was then used as a program. In such cases we usually want the compiled program to be such that any machine that computes the compiled program (under f_R) thereby computes the original program (under some specified f_R'). This is a standard account (see, e.g., Clark & Cowell 1976). If a grammar were part of a program that was compiled, it would clearly be represented as data for the compiler. In fact, **Samet** and **Berwick** propose a view much like this – a view originally proposed in Fodor, Bever & Garrett 1974 – in which the grammar is used as data by a program that generates a corresponding recognition algorithm. This view is committed to a representation of the grammar that is used as data *and* to a representation of the recognition algorithm, and so it is a theory committed to the hypothesis Hb which I criticize.

The present framework can similarly provide the standard account of interpreters, and so I cannot see why **Lipton** and **Davis** worry about them. An interpreter can be seen, as Lipton suggests, as allowing a program in a high-level language like LISP to be *used* by a machine that uses a lower-level machine language, and the interpreter can also be seen as a program that takes the high-level language as data. Again, it is no problem for the present account that in such cases an encoding can be treated both as data and as a program.

Hardwired computation. A hardwired circuit can, as **Berwick** points out, simulate the computation of any program; and Berwick suggests that the mapping between parts of the circuits and parts of the programs which is exploited to get the results I mention in note 2 can be taken as defining a realization function. He is right that if this were the case, my distinction between program computation and program use would collapse. The fact is, though, that a piece of circuit that computes some function cannot ipso facto be taken as a representation of some syntactic object that denotes that function. Which syntactic object would it be? Furthermore, these states do not govern the system in the way that an encoding of a program governs a program-using system, as I pointed out above in answer to **Harman**'s similar idea.

Underdetermination. My arguments are seen by **Samet** and **Lipton** as showing only that H3 is underdetermined by the data, and since even the best-supported scientific theories are underdetermined, my conclusion that H3 is not supported by available evidence does not follow. That is, they see me as pointing out that for any hypothesis like H3 we can find another hypothesis which cannot be ruled out on the basis of available data. To argue from this premise to my conclusion would obviously be fallacious. What I do instead is point out that not only is all the evidence adduced in support of H3 accounted for by Hh, but Hh is really the better hypothesis. In the first place, it is simpler. H3 just posits a more complicated language-processing mechanism than Hh does. In the second place, the lack of plasticity of grammatical knowledge is predicted by Hh but not by H3. And in the third place, mechanisms of the sort posited by Hh tend to be more efficient than mechanisms of the sort posited by H3, in which the computation is mediated by accessing a representation of the program computed. This point favors Hh, since human language processing shows signs of impressive efficiency: it is very fast and can be carried out at the same time other sophisticated information-processing tasks are underway. Since I not only argue that there is no reason to prefer H3 or Hb in the light of any evidence but also point out these advantages in Hh, my argument is much stronger than these commentators allow.

The evidence for H3. Other commentators suggest that I may have overlooked some evidence which actually does favor a representational hypothesis like H3 or Hb. **Berwick** argues that the assumption that the "move α" rule is literally embedded and causally engaged in linguistic processing is supported by its ability to explain such things as parsing efficiency, but he does not provide any reason for thinking that this proposal does not suffer from the same problems that I point out for the (now widely rejected) passive transformation. Move α specifies a mapping between trees, and this mapping can be computed without accessing any representation of that rule. **Lockman** suggests that I underestimate the force of the evidence from language acquisition. He seems to think that any system which computes one function at one time and another function at another time must be a program-using system. This does not seem right. As I point out, "plasticity" provides an argument for a representational hypothesis not because "hardwired" systems are utterly unable to change but because it is a plausible empirical assumption that their operation cannot change radically in a manner appropriate to the stimulus in very short times. See Pylyshyn's (1980) discussion of plasticity for a more detailed discussion of a similar view. As **Marshall** suggests, and as Pylyshyn (1980, forthcoming) has argued, there are many examples of real "plasticity," and I do take these as supporting representational claims. I do not take them as providing good support for a third-level claim that some program is governing the computation. **Dennett** and **Marshall** wonder what would support such a claim, and here I think Lockman is right in suggesting that support for such a view will probably not be available until we have a much better idea of the basic processing abilities, the "functional architecture" that is flexibly employed in the execution of cognitive tasks. The moral of Pylyshyn's work, as I read it, is that it is at present dreadfully hard to put your finger on even a single piece of that architecture (even granting **Morton**'s point that we do not need a *complete* first-level theory before we can begin developing second- and even third-level theories of isolable parts of the computational system).

The significance of the conclusion. Aside from considering my argument flawed, **Lipton** suggests that nothing hinges on the truth of my conclusions: all of this is a tempest in a teapot. **Gross** cannot see how any linguists could derive any insight from my conclusions. **May** says that my arguments leave linguistic methodology untouched. **Thompson** says that I have contributed nothing except my scepticism. And **Berwick** notes that since we can actually design circuits to compute any program directly, and vice versa, we should not be particularly concerned with whether the representational views are correct or not. I disagree. If it is true that linguistic theory does not tell us whether the grammar is encoded or not, and if it is true that it does not specify the language-processing algorithm, then theories of language processing must stand on their own strengths since they will not follow from linguistic theory, and, on the other hand, linguists do not need to worry about the details of language-processing theories since their own work is not implicated. As **Demopoulos & Matthews** point out, the issues here are important because some linguists reject this view and consequently reject some linguistic theories on the grounds that they do not provide a realistic view of language processing (Bresnan 1978; Kaplan & Bresnan, in press b). I think that it is the simplistic idea that grammars are actually represented and used in computations eventuating in linguistic behavior that has engendered this view. If my arguments are correct, they show that evidence adduced in support of most linguistic theories does *not* have a direct bearing on theories like H3. I did not explore the relation between the evidence that motivates linguistic theories and H2, but it is clearly going to be similarly remote; it is far from clear what, exactly, the relation is (see Berwick & Weinberg, 1983a, for a good discussion of this topic). Thus the simplistic idea is not supported by the evidence, and linguists should recognize that the proposal that they should require their theories to have a direct bearing on computational accounts of human language processing marks a significant departure from the tradition.

Another construal of the mental-representation hypothesis. If H3 and Hb are not supported by any of the evidence offered by linguists in support of M, and if evidence relevant to H3 and Hb has been ignored, as I suggest, it is hard to believe that linguists intended any computational construal of M. The problem then is: what do they mean by M? **Demopoulos & Matthews, Harman,** and **Botha** suggest that linguistic theory is not committed to anything like Hb and that it is a mistake to slip into any computational construal of the mentalistic claims of linguistic theory.

As **Demopoulos & Matthews** point out, to assume that M can be restated in computational terms is to suppose that hypotheses expressed in a mentalist vocabulary are reducible in a very simple manner to hypotheses expressed in a computational vocabulary. In short, it is to accept a strong reductionist assumption. What my paper presents, then, is an argument that a "reduced" version of M not only is unsupported by the evidence but also appears to be rather different from anything that linguists seem to have intended. **Harman, Botha** and **Demopoulos & Matthews** argue that we should reject the reductionist assumption and keep to the mentalist idiom; then we can see that M is both well supported by the adduced evidence and compatible with the linguists' intentions. The point is that although my argument bears on various computational theories of language processing, it raises no problem for the mental-representation hypothesis M if we reject the reductionist assumption and grant the following points: first, the grammar of a language characterizes the knowledge of a speaker–hearer; second, all knowledge is mentally represented; and finally, mentally represented knowledge is used to govern behavior. I agree with these commentators about the logic of the situation. The point to note is that this move obviously will be unappealing if the reductionist assumption is independently supported as Fodor (1975; 1980; 1981a) and others have argued. Rejecting the reductionist assumption divides mentalistic psychology from computational accounts in a way that will have serious implications for psychological theories in which the two sorts of account are merged. As Fodor points out (see, e.g., 1980, p. 63), this merging is so pervasive as to make it the distinguishing characteristic of contemporary cognitive psychology. Furthermore, the fact that a pure mentalist theory is true of humans is something that will call for explanation, and it is plausible that this explanation will be cast in computational terms. The computational theory will still face the problem of assessing the computational significance of M.

References

Aho, A. V. & Ullman, J. D. (1972) *The theory of parsing, translation, and compiling.* Prentice-Hall. [RW]

Berwick, R. (1982) *Locality principles and the acquisition of syntactic knowledge.* Ph.D. thesis, MIT Department of Computer Science and Electrical Engineering. [RCB]

Berwick, R. C. & Weinberg, A. S. (1983a) The role of grammars in theories of language use. *Cognition* 13:1–61. [tarEPS]

(1983b) *The grammatical basis of linguistic performance.* MIT Press. Forthcoming. [RCB]

Botha, R. B. (1980) Methodological bases of a progressive mentalism. *Synthese* 44:1–112. [RPB]

(1982) On Chomskyan mentalism: A reply to Peter Slezak. *Synthese* 53:123–41. [RPB]

Bratley, P., Dewar, H. & Thorne, J. P. (1967) Recognition of syntactic structure by computer. *Nature* 216:969–73. [JCM]

Bresnan, J. (1978) A realistic transformational grammar. In: *Linguistic theory and psychological reality,* ed. J. Bresnan, M. Halle, & G. Miller. MIT Press. [rEPS]

Bresnan, J. W. & Kaplan, R. M. (1982a) Lexical-functional grammar: A formal system for grammatical representation. In: *The mental representation of grammatical relations,* ed. J. W. Bresnan, MIT Press. [HT]

(1982b) Grammars as mental representations of language. In: *The mental representation of grammatical relations,* ed. J. W. Bresnan. MIT Press. [WD]

Burks, A. W. (1963) Programming and theory of automata. In: *Computer programming and formal systems,* ed. P. Braffort & D. Hirschberg. North-Holland. [rEPS]

Carr, T. H., Davidson, B. J. & Hawkins, H. L. (1978) Perceptual flexibility in word recognition: Strategies affect orthographic computation but not lexical access. *Journal of Experimental Psychology: Human Perception and Performance* 4:674–90. [JCM]

Chomsky, N. (1965) *Aspects of the theory of syntax.* MIT Press. [JDM, taEPS]

(1969) Comments on Harman's reply. In: *Language and philosophy,* ed. S. Hook. New York University Press. [tarEPS]

(1972) *Language and mind.* New York: Harcourt Brace Jovanovich. [taEPS]

(1973) Conditions on transformations. In: *A festschrift for Morris Halle,* ed. S. R. Anderson & P. Kiparsky. Holt, Rinehart and Winston. [taEPS]

(1975) *Reflections on language.* Pantheon Books. [taEPS]

(1977) On Wh-movement. In: *Formal syntax,* ed. P. W. Culicover, T. Wasow & A. Akmajian. Academic Press. [taEPS]

(1978) On the biological basis of language capacities. In: *Psychology and biology of language and thought: Essays in honor of Eric Lenneberg,* ed. G. A. Miller and E. Lenneberg, pp. 199–220. Academic Press. [JCM]

(1980a) *Rules and representations.* Columbia University Press. [RPB, WD, taEPS]

(1980b) Rules and representations. *Behavioral and Brain Sciences* 3:1–61. [RMH, tarEPS]

(1980c) On binding. *Linguistic Inquiry* 11:1–46. [taEPS]

(1981) *Lectures on government and binding.* Foris Publications. [RPB, GH]

(1982) *Some concepts and consequences of the theory of government and binding.* MIT Press. [GH]

Chomsky, N. & Lasnik, H. (1977) Filters and control. *Linguistic Inquiry* 8:425–504. [JDM]

Chomsky, N. & Miller, G. A. (1964) Formal properties of grammars. In: *Handbook of mathematical psychology,* vol. 1, ed. R. D. Luce, R. R. Bush, & E. Galanter, pp. 323–418. John Wiley & Sons Inc. [MG]

Clark, K. & Cowell, D. (1976) *Programs, machines, and computation: An introduction to the theory of computing.* McGraw-Hill. [rEPS]

Clark, K. L. & McCabe, F. G. (1979) The control facilities of IC-PROLOG. In: *Expert systems in the micro electronic age,* ed. D. Mitchie. Edinburgh University Press. [rEPS]

Crick, F. H. C. (1979) Thinking about the brain. *Scientific American* 241:181–88. [MG]

Davis, M. (1978) What is a computation? In: *Mathematics today: Twelve informal essays,* ed. L. A. Steen, pp. 241–67. Springer-Verlag. [MD]

Davis, M., ed. (1965) *The undecidable.* Raven Press. [MD]

Field, H. H. (1978) Mental representations. *Erkenntnis* 13:9–61. [taEPS]

Fodor, J. A. (1975) *The language of thought.* Crowell. [tarEPS]

(1980) Methodological solipsism considered as a research strategy in cognitive psychology. *Behavioral and Brain Sciences* 3:63–109. [rEPS]

(1981a) *Representations.* Bradford Books. [WD, tarEPS]

(1981b) Some notes on what Linguistics is about. In: *Readings in the philosophy of psychology,* volume 2, ed. N. Block, pp. 197–207. Harvard University Press. [WD]

(1981c) The present status of the innateness controversy. In: *Representations.* Bradford Books. [RMH]

(1983) *The modular theory of mind: An essay on faculty psychology.* Bradford Books. [WD, taEPS]

Fodor, J. A., Bever, T. G. & Garrett, M. F. (1974) *The psychology of language.* McGraw-Hill. [RCB, tarEPS]

Fodor, J. A. & Pylyshyn, Z. W. (1981) How direct is visual perception? *Cognition* 9:136–96. [taEPS]

Gazdar, G. (1982) Phrase structure grammar. In: *The nature of syntactic representation,* ed. P. Jacobson & G. K. Pullum. Reidel. [rEPS, HT]

Grimson, W. E. L. (1981) A computer implementation of a theory of human stereo vision. *Philosophical Transactions of the Royal Society of London* B292:217–53. [JCM]

Halle, M. & Stevens, K. (1962) Speech recognition: A model and a program for research. *IRE Transactions on Information Theory, IT-8,* pp. 155–59. [taEPS]

Harman, G. (1967) Psychological aspects of the theory of syntax. *Journal of Philosophy* 64:75–87. [taEPS]

(1969) Linguistic competence and empiricism. In: *Language and philosophy,* ed. S. Hook. New York University Press. [taEPS]

Hawkins, H. L., Reicher, G. M., Rogers, M. & Peterson, L. (1976) Flexible coding in word recognition. *Journal of Experimental Psychology: Human Perception and Performance* 2:380–85. [JCM]

Hill, F. & Peterson, G. (1981) *Introduction to switching theory and logical design*. Wiley. [RMH]

Hubel, D. H. (1979) The brain. *Scientific American* 241(3):44–53. [taEPS]

Johnson-Laird, P. N. (1977) Procedural semantics. *Cognition* 5:189–214. [taEPS]

Jonides, J. & Gleitman, H. (1972) A conceptual category effect in visual search: O as letter or as digit. *Perception and Psychophysics* 12:457–60. [JCM]

Joshi, A. K., Levy, L. S., & Takahashi, M. (1975) Tree adjunct grammars. *Journal of Computer and System Sciences*, pp. 136–63. [MG]

Julesz, B. (1971) *Foundations of cyclopean perception*. University of Chicago Press. [taEPS]

Kaplan, R. & Bresnan, J. (in press a) A formal system for grammatical representation. In: *The mental representation of grammatical relations*, ed. J. Bresnan. MIT Press. [taEPS]

(in press b) Grammars as mental representations of language. In: *The mental representation of grammatical relations*, ed. J. Bresnan. MIT Press. [rEPS]

Katz, J. J. (1966) *The philosophy of language*. Harper and Row. [taEPS]

Keyser, S. J. & Pinker, S. (1980) Direct vs. representational views of cognition. *The Behavioral and Brain Sciences* 3:389–90. [rEPS]

Kuno, S. & Oettinger, A. G. (1963) Multiple-path syntactic analyzer. *Information Processing 1962*, North-Holland. [HT]

Lakoff, G. & Thompson, H. (1975) Introducing cognitive grammar. *Proceedings of the first annual meeting*, ed. C. Cosen et al., pp. 295–313. Berkeley Linguistics Society. [JDM, HT]

Laudan, L. (1977) *Progress and its problems*. University of California Press. [JDM]

Lewontin, R. C. (1983) The liberation of biology. *New York Review of Books*, Jan. 20. [TR]

Liberman, A. M., Cooper, F. S., Shankweiler, D. P. & Studdert-Kennedy, M. (1967) Perception of the speech code. *Psychological Review* 74:431–61. [rEPS]

Lycan, W. G. (1981) Form, function and feel. *The Journal of Philosophy* 78:24–50. [taEPS]

McCawley, J. D. (1982) How far can you trust a linguist? In: *Language, mind, and brain*, ed. Thomas Simon and Robert Scholes, pp. 75–87. Albex. [JDM]

Macnamara, J. & Kushnir, S. (1971) Linguistic independence of bilinguals: The input switch. *Journal of Verbal Learning and Verbal Behavior* 10:480–87. [JCM]

Marr, D. (1979) Representing and computing visual information. In: *Artificial intelligence: An MIT perspective, volume 2*, ed. P. H. Winston & R. H. Brown. MIT Press. [taEPS]

(1980) Visual information processing: The structure and creation of visual representations. *Philosophical Transactions of the Royal Society of London* B290:199–218. [JCM]

Marr, D. & Poggio, T. (1976) From understanding computation to understanding neural circuitry. *MIT Artificial Intelligence Memo 357*, Cambridge, Mass. [JM, taEPS]

Marr, D. & Poggio, T. (1979) A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B204:301–28. [JCM]

Marshall, J. C. (1980) On the biology of language acquisition. In: *Biological studies of mental processes*, ed. D. Caplan, pp. 106–48. MIT Press. [JCM]

Matthews, R. J. (1982) Knowledge of language in a theory of language processing. Presented at the conference "Constraints on modelling real-time processes," sponsored by the Max Planck Institute for Psycholinguistics, Nijmegen, and the Center for Psychosocial Research, Chicago, and held in Saint-Maximin, France, June, 1982. [taEPS]

Mehler, J., Morton, J. & Jusczyk, P. (submitted for publication) On reducing language to biology. [JM]

Miller, G. A. & Chomsky, N. (1963) Finitary models of language users. In: *Handbook of mathematical psychology*, volume 2, ed. R. D. Luce, R. Bush & E. Galanter. Wiley. [taEPS]

Pereira, F. & Warren, D. H. D. (1980) Definite clause grammars for language analysis: A survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence* 13:231–78. [rEPS, HT]

Pinker, S. (in press) A theory of the acquisition of lexical-interpretive grammars. In: *The mental representation of grammatical relations*, ed. J. Bresnan. MIT Press. [taEPS]

Pippenger, N. & Fischer, M. J. (1979) Relations among complexity measures. *Journal of the ACM* 26:361–81. [taEPS]

Putnam, H. (1967) Psychological predicates. In: *Art, mind and religion*, ed. W. H. Capitan & D. D. Merrill. University of Pittsburgh Press. [rEPS]

Pylyshyn, Z. (1980) Computation and cognition: Issues in the foundation of cognitive science. *The Behavioral and Brain Sciences* 3:11–169. [WD, RMH, tarEPS]

(forthcoming) *Computation and cognition*. Bradford. [tarEPS]

Schank, R. (1977) Response to Dresher and Hornstein. *Cognition* 5:133–45. [RW]

Schnorr, C. P. (1976) The network complexity and Turing machine complexity of finite functions. *Acta Informatica* 7:95–107. [taEPS]

Searle, J. R. (1980) Rules and causation. (Commentary on Chomsky, 1980b.) *The Behavioral and Brain Sciences* 3:37–38. [WD, taEPS]

Shulman, H. G. & Davison, T. C. B. (1977) Control properties of semantic coding in a lexical decision task. *Journal of Verbal Learning and Verbal Behavior* 16:91–98. [JCM]

Stabler, E. P. (1981) *Issues in the foundations of cognitive psychology*. Unpublished Ph.D. thesis, MIT, Cambridge, Massachusetts. [taEPS]

(1982) Database and theorem prover designs for question answering systems. Cogmem No. 12, Centre for Cognitive Science Technical Memorandum Series, The University of Western Ontario. [rEPS]

Stabler, E. P. & Elcock, E. W. (1983) Knowledge representation in an efficient deductive inference system. Submitted for presentation at the Logic Programming Workshop '83, Aldeia des Acoteias, Portugal. [rEPS]

Stelmach, G. E., ed. (1978) *Information processing in motor control and learning*. Academic Press. [taEPS]

Stoy, J. E. (1977) *Denotational semantics: The Scott–Strachey approach to programming language theory*. MIT Press. [taEPS]

Swinney, D. A. (1981) Lexical processing during sentence comprehension: Effects of higher order constraints and implications for representation. In: *The cognitive representation of speech*, ed. T. Myers, J. Laver, and J. Anderson, pp. 201–09. North-Holland. [JCM]

Szentagothai, J. & Arbib, M. A. (1975) *Conceptual models of neural organization*. MIT Press. [taEPS]

Tanenhaus, M. K., Leiman, J. M. & Seidenberg, M. S. (1979) Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior* 18:427–40. [JCM]

Tennant, R. D. (1976) The denotational semantics of programming languages. *Communications of the ACM* 19:437–53. [taEPS]

Ullman, S. (1980) Against direct pereception. *The Behavioral and Brain Sciences* 3:373–81. [rEPS]

Van Emden, M. H. & Kowalski, R. A. (1976) The semantics of predicate logic as a programming language. *Journal of the Association for Computing Machinery* 23:733–42. [rEPS]

Von Neumann, J. (1958) *The computer and the brain*. Yale University Press. [taEPS]

Webb, Judson C. (1980) *Mechanism, mentalism, and metamathematics*. D. Reidel. [MD]

Young, J. Z. (1964) *A model of the brain*. Clarendon Press. [taEPS]