What's wrong with Grandma's guide to procedural semantics: A reply to Jerry Fodor*[1]

P. N. JOHNSON-LAIRD
Laboratory of Experimental Psychology
University of Sussex

"Procedural semantics" is a label for a loose confederation of theories of meaning that rely on an analogy between ordinary language and high-level programming languages: compiling and executing a program correspond rather naturally to stages in a person's comprehension of an utterance. The analogy has been most strikingly exploited by workers in artificial intelligence, who have devised a variety of programs that manipulate natural language (see e.g., Winograd, 1971; Woods, 1973; Schank, 1972; Longuet-Higgins, 1972; Davies and Isard, 1972). But Miller and Johnson-Laird (1976) have also adopted a procedural approach to the study of the mental lexicon, and have argued that it seems particularly suitable for developing psychological theories of the comprehension and production of discourse (see Johnson-Laird, 1977a).

The whole enterprise is attacked by Jerry Fodor (1978) in "Tom Swift and his Procedural Grandmother", a critique that is a volatile mixture of the theoretical, the rhetorical, and the hobby-horsical. The theoretical remarks are disputable. The rhetoric is amusingly disputatious. There is no disputing against hobby-horses.

The gist of Fodor's case runs as follows: *Procedural semantics is parasitic on the classical model-theoretic semantics of truth, reference and modality. Procedural semantics attempts to interpret English by translating (i.e., compiling) it into the machine language of a computer; but for this operation to provide a true semantic theory, the machine language has to have a classical*

*Requests for reprints should be sent to: P. N. Johnson-Laird, Centre for Research on Perception and Cognition, Laboratory of Experimental Psychology, University of Sussex, Brighton, BN1 9QG, England.

[1] I am indebted to Jerry Fodor for sending me successive versions of his critique of procedural semantics, and for at all times conducting himself according to the Marquis of Queensbury Rules. I am also grateful to Jacques Mehler for his dispassionate refereeing. Likewise, I must acknowledge with thanks the untiring efforts of those who have tried to coach me in the intricacies of procedural and model-theoretic semantics: Steve Isard, Christopher Longuet-Higgins, George Miller, Mark Steedman, and Bill Woods. Steve Isard very kindly made available to me his own unpublished reply to an earlier version of Jerry Fodor's paper, from which I have borrowed more than I can acknowledge. Finally, I am most grateful to Stuart Sutherland, my 'second' in this matter, who has striven to ensure that I strike my blows in good English. My research is supported by a grant from the Social Science Research Council (Great Britain).

0010-0277/78/0006-0249$2.25 © Elsevier Sequoia S.A., Lausanne Printed in the Netherlands

interpretation. Moreover, the interpretation assigned a programming language sentence by the compiler is not, normally, its intended interpretation: computers do not know or care what the programs they run are about. And, because machine language is interpreted solely for machine states and processes, there is nothing available in procedural semantics to reconstruct the classical relation of reference that holds between, say, the term 'Lucy' and the individual, Lucy. Nevertheless proceduralists widely suppose that English can be translated into a machine language provided that it is enriched with the names for the states of sensory transducers. This assumption resurrects the discredited empiricist thesis that concepts (other than logical ones) can be reduced by definitions to expressions in a language of sensations, and that percepts are likewise constructed from check-lists of sensory features. It also leads directly to a form of verificationism, the equally discredited doctrine that only logical truths and empirically testable sentences are meaningful. Proceduralists either subscribe to verificationism or else reject it at the cost of having an incomplete semantic theory. In fact, a natural language such as English cannot be reduced at all: its vocabulary is probably not much larger than it needs to be given the expressive power of the language. Finally, procedural semantics confuses semantic theories with theories of sentence comprehension, but even here its pay-off has been negligible.

I intend to show that each of these assertions is either false or else irrelevant to the proper evaluation of procedural semantics. But, since they are largely of a philosophical nature, and almost entirely innocent of empirical consequences, my main aim in this reply is, not to re-interpret psychological phenomena, but to locate and to elucidate the errors in Fodor's Guide to Procedural Semantics.

Procedural semantics is parasitic on the classical model-theoretic semantics of truth, reference and modality. A major thrust of Fodor's paper is that procedural semantics is intended as an alternative to classical semantics, the tradition initiated by Frege and brought to fruition in model-theoretic accounts of meaning. Such a semantics for a language is set up by replacing reality with a *model*, and by postulating an *interpretation* that connects the language to the model. An interpretation is essentially a function that for each individual constant in the language picks out the corresponding individual in the model, and for each predicate in the language, picks out the set of individuals in the model (or set of ordered sets of individuals where the predicate takes several arguments) that satisfy it. The interpretation function also operates recursively to define the truth or falsity of sentences in terms of such interpretations of their constituents. A model-theoretic semantics for a (fragment of) natural language usually arranges for this recursive machinery to work in parallel with the rules of syntax for the language, and makes use of a model containing a set of "possible worlds" (i.e., possible states of affairs) in order to interpret modal sentences, and other sentences of a similar sort. The meaning (or intension) of a sentence is accordingly a function that maps the possible worlds onto a truth value. The reference (or extension) of the sentence is its truth value in the particular world that obtains. There can be many different models for a given language: logicians are primarily concerned not with which is the right model but with principles that hold over all models.

Despite Fodor's claim, procedural semantics is not intended to supplant classical model-theoretic semantics. Such a proposal would be misguided as can be shown by a simple example. Suppose a psychologist proposes a procedural model of how people reason with propositions. Fodor arrives on the scene and points out that the theory is parasitic upon the model-theoretic semantics for the propositional calculus, that is, the classical apparatus of truth tables. The claim may well be true, depending on what he means by "parasitic"; but it is irrelevant. The classical theory has no implications for the mental processes by which people reason: it simply specifies what counts as a valid deduction. Since the psychologist is interested in the particular system of cognitive operations that people employ to make valid deductions, and since it is an empirical task to discover what that system is, model–theoretic approaches offer no solutions to his problems (Johnson-Laird, 1977a, p. 193). The two sorts of theory are not in competition.

Procedural semantics attempts to interpret English by translating it into a classically interpreted machine language. The only tactful answer to this assertion is to whistle half a dozen bars of Lillabullero. It is simply not true. But, since the misapprehension lies at the heart of Fodor's conception of procedural semantics, it should be instructive to examine it in more detail.

Fodor starts from three quotations (to which I have restored some of the material that he omits):

> These artificial languages, which are used to communicate programs of instructions to computers, have both a syntax and a semantics. Their syntax consists of rules for writing well-formed programs that a computer can interpret and execute. The semantics consists of the procedures that the computer is instructed to execute. If, for example, a programming language permits an instruction like: x and y →, it might mean that the computer is to add the values of x and y, and to print the result.
>
> ...we might speak of the intension of a program as the procedure that is executed when the program is run, and of the extension of a program as the result the program returns when it has been executed.
>
> Johnson-Laird (1977)

> We can call the model of semantics used in our system the "procedure model". The primary organisation of knowledge is in a deductive program with the power to combine information about the parsing of the sentence, the dictionary meannings of its words, and non-linguistic facts about the subject being discussed. Any relevant bit of knowledge can itself be in the form of a program or procedure to be activated at an appropriate time in the process of understanding. The program operates on a sentence to produce a representation of its meaning in some internal language, in our case PLANNER.
>
> Winograd (1971, p. 409)

With no further justification whatsoever, Fodor takes for granted that proceduralists propose that the meaning of a sentence is its representation in machine language. The compiler, as Fodor puts it, is the semantic theory, and semantic interpretation consists in translating a sentence into machine language, which in turn should receive a classical semantic interpretation. The equivocation here is subtle, but disastrous. Neither Winograd nor Johnson-Laird spoke of expressions in machine language as representing meanings. Indeed, as Steve Isard has pointed out in an unpublished reply to Fodor, to define the semantics of a high-level language by its compiler is analogous to defining the semantics of English in terms of neural activity. It is blatant reductionism of the sort that elsewhere receives Fodor's justified scorn. One of the great virtues of a digital computer is precisely that it is a working illustration of the futility of reductionism: a program in a high-level language such as PLANNER concerns goals, objects, and properties, not patterns of bits and storage locations. The organisation of the program has a functional autonomy that is totally independent both of the particular machine language into which it is ultimately compiled and of the physics of the particular computer that runs the program. It is not my aim to teach Fodor's procedural grandmother to suck eggs, or to tell Fodor something that he already knows, but such arguments are hardly novel (cf., Fodor, 1968; Putnam, 1975).

Fodor argues that if proceduralists do not think English can be paraphrased in some sort of machine language, then they owe the world some alternative account of how a programming language can be classically interpreted. In fact, this alternative exists and a brief inquiry into it delivers the *coup de grâce* to Fodor's debilitated notion of procedural semantics. The pioneers of model-theoretic semantics for programming languages, Scott and Strachey (1971), write as follows: "Compilers of high-level languages are generally constructed to give the complete translation of the programs into machine language. As machines merely juggle bit patterns, the concepts of the original language may be lost or at least obscured during this passage. The purpose of mathematical [i.e., model-theoretic] semantics is to give a correct and meaningful correspondence between programs and mathematical entities in a way that is entirely independent of an implementation". To take a specific example, a list-processing language will contain an instruction that returns the *head* of a list, say, HD(x), which, if x is the list [Tom Dick Harry], returns "Tom" as its value when it is executed. If you wish to characterise the meaning of this instruction in terms of a model-theoretic semantics, then you would certainly not do so by relating HD(x) first to an expression in machine language and then interpreting this expression. Such a tactic would plainly necessitate a different semantics for each of the many different machine languages into which the list-processing language could be translated. What you would do, following in the steps of Scott and Strachey, would be to set up a direct interpretation of the list-processing language, relating it to an abstract model containing lists, their elements, and various functions. It does not matter how a machine actually represents 'HD(x)', lists, or their constituents, provided that it does so in a way that is in accordance with their semantics.

Unfortunately, Fodor's error here is so egregious that it largely wrecks the rest of his paper. But, since he does raise some other interesting points, I shall refrain from the *argumentum fistulatorium* and persevere with my reply.

The interpretation assigned a programming language sentence by the compiler is not, normally, its intended interpretation. What Fodor has in mind here is that "machines typically don't know (or care) what the programs that they run are about; all they know (or care about) is how to run their programs". Fodor illustrates the point by considering two programs: one simulates the Six Day War and the other simulates a game of chess, and they just happen to be indistinguishable when they are compiled. Of course, exactly the same example can be created in terms of a model-theoretic semantics, and, in fact, the same mathematical language is often interpreted using different models. Indeed, the case can even arise in natural language. Consider a *description* of the Six Day War in which the various deployments of forces are identified by different codewords, so that "pawns attack knight" actually means that Israeli infantry surround a tank corps. Now, it just so happens (by parity with Fodor's example) that a description of the Six Day War constitutes a description of a game of chess. What moral should we draw? That a speaker of English could find out what was being referred to, but a computer could not? I see no reason to suppose that computers cannot in principle be programmed to deal with such ambiguities.

What Fodor seems to have lost sight of in his example is the distinction between a theory and what the theory is about. All theories are abstractions. It would be silly to criticize a theory of X-rays on the grounds that it, the theory, was not radio-active. Likewise, it seems silly to criticize a computer program that simulates the Six Day War on the grounds that the machine does not know its intended interpretation, and silly to criticize a procedural or model-theoretic semantics on the grounds that it does not *know* (or care about) the intended interpretations that its models.

Nothing is available in procedural semantics to reconstruct such classical semantic relations as the one that holds between 'Lucy' and Lucy, or between 'chocolate cake' and chocolate cakes. Alas, classical semantics does not reconstruct the relation of reference that holds between expressions such as 'Lucy' and entities such as Lucy; it merely postulates a primitive and unanalyzed function that maps terms in the language onto entities in the model. Indeed, so little is it concerned with such matters that mathematicians making use of model-theoretic semantics often do not bother to distinguish between individual constants in the language and the individuals in the model (see Robin, 1969). Fodor's criticism accordingly applies to semantic theories in general, as he himself concedes rather later in the paper: "it is, of course, not very interesting to say that 'chair' refers to chairs, since we have no *theory* of reference and we have no *mechanism* to realise the theory".

Proceduralists suppose that English can be translated into an ENRICHED machine language. After strenuously arguing against the notion that translation into machine language provides a satisfactory semantics, a notion that he has unjustifiably attributed to proceduralists, Fodor finally concedes that no one was ever likely to have held such a view in the first place. What is widely supposed, he suggests, is that English can be translated into an *enriched* machine language. He has in mind equipping a computer with sensory transducers and a machine language with names for their input (sic) and output states. Many researchers in Artificial Intelligence believe (or had better believe), he says, that English can be translated into this MLT, or Machine Language enriched with nourishing Transducer-state names. But, this doctrine is nothing other than a resurrected version of the empiricist principle that "*every* non-logical concept is reducible to sensation concepts (or, in trendier versions, to sensation plus motor concepts) via coordinating definitions".

Fodor erred in assuming that procedural semanticists aim to translate English into machine language (enriched or otherwise). His thesis that many workers in AI are committed to a *sensation* language of the sort envisaged by John Locke is an extraordinary *faux pas.* If the work on scene analysis has a patron philosopher, it is undoubtedly Immanuel Kant. Here is not the place to review these studies, but it is relevant to point out that one of the clearest morals to have emerged from them is that any simple empiricist program that attempts to build up percepts from the properties of the sensory input *alone* is unworkable. Programs require a knowledge of a variety of domains from the projective geometry of three-dimensional objects (e.g., Waltz, 1975) to the prototypical shapes of certain sorts of object (e.g., Marr and Nishihara, 1976). Moreover, there is no good reason to suppose that the language of sensations corresponds to the output of sensory transducers: "sensations are not psychic atoms in perceptual compounds; they are abstracted from percepts by a highly skilled act of attention" (Miller and Johnson-Laird, 1976, p. 29). Only a theorist committed to a one-to-one relation between mental language and natural language is likely to identify the output of sensory transducers with the language of sensations.

Fodor's misapprehension that vision programs are exercises in naive empiricism leads him astray on what procedural semantics has to say about the relations between language and perception. In his view, proceduralists represent the meaning of a term such as 'chair' by relying on the sensory predicates with which they have enriched machine language. Hence, the meaning of 'chair' is simply a set of *sensory* properties: semantic decomposition of the lexicon parallels sensory decomposition of percepts. This thesis is false. Miller and Johnson-Laird (1976, Sec. 4) in fact argue that many aspects of the meaning of words have no perceptual correlates; that where there is such a link it is mediated by a complex conceptual apparatus; and that a perceptual paradigm for an object is not a set of *sensory* properties.

Fodor's diatribe against empiricist theories of perception – they are remarkably old news, they are reductionist, they are atomistic, and so on and on – seems otherwise correct, but irrelevant to procedural semantics.

Proceduralists either subscribe to verificationism or else reject it at the cost of having an incomplete semantic theory. Fodor's arguments about empiricism are perhaps tendentious in that they plainly lead up to the heart of his criticism, namely, that procedural semantics is in a dilemma about verificationism. (In Section 1 of his paper, he claims that procedural semantics is "an archaic and wildly implausible form of verificationism". Later, he relents and allows that some proceduralists are not verificationists. I shall deal with this second line of argument on the grounds that it is nearer to the truth.)

The basic doctrine of verificationism is that a sentence is meaningful, as opposed to meaningless, only if its truth or falsity (or probability to some degree) can in principle be established empirically. This view is primarily associated with the Logical Positivists, who sometimes went even further and identified the meaning of a sentence with its method of verification, a central tenet of operationalism. Obviously, the doctrine was not intended to apply to analytic or necessarily true sentences.

The dilemma that Fodor poses for procedural semantics is whether or not to embrace verificationism. He quotes Woods (1975, p. 39):

> In order for an intelligent entity to know the meaning of such sentences [as "Snow is white"] it must be the case that it has stored somehow an effective set

> of criteria for deciding in a given possible world whether such a sentence is true or false.

And this view, he says, is about the strongest form of verificationism that anyone has ever endorsed: it implies that merely in virtue of having learned English, a speaker possesses an algorithm for determining the truth value of such sentences as: "God exists", "positrons are made of quarks", "Aristotle liked onions". Unfortunately, Fodor has overlooked a qualification that Woods (1975, p. 40) makes on the very next page:

> The case presented above is a gross oversimplification of what is actually required for an adequate procedural specification of the semantics of natural language. There are strong reasons which dictate that the best one can expect to have is a partial function which assigns true in some cases, false in some cases, and fails to assign either true of false in others. There are also cases where the procedures require historical data which is not normally available and therefore cannot be directly executed.

The reader is referred to Woods (1973, 1978) for a further discussion of his actual views. Since no proceduralist appears to be directly impaled on this point of Fodor's arguments, let us turn to the other horn of the dilemma.

According to Fodor, the grounds on which Miller and Johnson-Laird (1976) deny that they are verificationists are obscure. He accordingly invents an account on their behalf, which has the consequence that their theory cannot give an interpretation for a term such as, "cause". Since this argument rests on Fodor's fallacy of the enriched machine language into which sentences are supposed to be interpreted, this horn is crumpled and impales no one. In fact, Miller and Johnson-Laird reject verificationism as a possible basis for a psychological theory of meaning on several grounds: verification is only one of a number of different conceptual operations that may be carried out as a consequence of understanding a sentence; and it runs into problems with the vagueness of ordinary language. The crux of their argument is that understanding a sentence is possible even when verification is not, e.g., "There's a gorilla in that closet whenever no one is trying to find out that there is". And, in their discussion of the process of verifying a claim such as "That is a book", they write:

> ...a meaning of the sentence must be clear before you undertake to verify it; if it were not, you would not know how to proceed with its verification. Understanding is antecedent to verification, not a consequence of verification. (Ibid. p. 126.)

Their theory of the meaning of a word is encapsulated in the following quotation:

> The meaning of 'book' is not the particular book that was designated, or a perception of that book, or the class of objects that 'book' can refer to, or a disposition to assent or dissent that some particular object is a book, or the speaker's intention (whatever it may have been), or the set of environmental conditions (whatever they may have been) that caused him to use this utterance, or a mental image (if any) of some book or other, or the set of other words associated with books, or a dictionary definition of 'book', or the program of operations (whatever they are) that people have learned to perform in order to verify that some object is conventionally labelled a book. We will argue that the meaning of 'book' depends on a general concept of books; to know the meaning is to be able to construct routines that involve the concept in an appropriate way, that is, routines that take advantage of the place 'book' occupies in an organised system of concepts. (Ibid. pp. 127 - 8.)

I hope that these remarks have dispelled any remaining obscurity as to why proceduralists are not necessarily committed to verificationism.

The vocabulary of a language like English is probably not much larger than it needs to be given the expressive power of the language. Fodor takes the view that the meanings of English words cannot be reduced to more elementary elements in some theoretical language. This is an interesting point of controversy, but not one that directly relates to procedural semantics. There are proceduralists who believe that the meanings of words can be decomposed into semantic primitives (Schank, 1972), but there are also proceduralists who take the contrary point of view (Winograd, 1974). I have dealt elsewhere with certain aspects of the controversy (Johnson-Laird, 1977b), but let me here take up one of Fodor's specific points without prejudice to the general issue of the viability of procedural semantics.

Fodor argues that "the vocabulary of a language like English is probably not much larger than it needs to be given the expressive power of the language", and that is why there are so few good examples of definitions of English words. (Spare a thought for Dr. Johnson, Noah Webster, and Sir James Murray, rotating like lathes in their graves.) In fact, the most immediate argument for the feasibility of semantic reduction is the existence of *Basic English* (Ogden, 1930): with a vocabulary of about 850 words, it is possible to say almost all of what one wants to say, and other words, if need be, can be defined in Basic English. Of course, a word accretes a variety of literary, historical, or scientific connotations, but if these expressive elements are excluded as irrelevant to truth conditions, then it is plain that English could be shorn of *defenestration, sexagenarian, privateer, triturate, eleemosynary, renitent, macarize, dilucidate, stiver, defluxion, abjudge, statutable, toise, argute, tritical, tabid, periapt, covin, iracund, obstipation, stridulous, gummous,* and thousands of other words, with no loss. Moreover, if you want to know what such words mean, or are in search of good definitions, then turn to the works of Johnson, Webster, and Murray, and their successors. Indeed, the meanings of many words can often be acquired only from dictionary definitions. Some words are easy to define, other words are extremely difficult to define without falling back on synonyms, ostension, examples of usage, or vicious circles (see Johnson-Laird and Quinn, 1976). This division of vocabulary appears to be one of the central features of the lexicon, and it is naturally accounted for by theories that decompose meanings into semantic primitives. Words that are close to expressing an unadorned primitive notion are hard to define, whereas words that express some combination of primitive notions are easy to define in terms of words for them. It is difficult to resist the conclusion that English contains a larger vocabulary than it strictly needs in order to meet the criteria of classical semantics (of which Fodor is so ardent an advocate).

Procedural semantics confuses semantic theories with theories of sentence comprehension, but even here its pay-off has been negligible. As Fodor makes abundantly clear, he demands (like the philosopher, Donald Davidson, 1967) that a semantic theory should relate language directly to the world. Proceduralists, claims Fodor, have made no contribution to such an account, but talk as though they had. It is perfectly true that procedural semantics is not an exercise in relating language directly to reality: what would the computer be doing but getting in the way, interposing itself between them, if it were? But, the claim that proceduralists forget they are in the business of establishing theories of internal representations strikes me as a contrived fiction – I hear the sound of coconut shells struck together as Grandmama rides off on her latest hobby-horse. In fact, many proceduralists doubt whether the Davidsonian philosophy for a semantics of natural language is feasible.

They are not alone. Certain philosophers have argued that linguistic expressions do not refer except in a derivative sense: it is people who refer, and they may do so by using linguistic expressions (see, e.g., Strawson, 1950; Austin, 1962; Searle, 1969). When Fodor talks of expressions as referring to objects and criticizes procedural semantics (along with all other theories) for failing to reconstruct this relation, he is perhaps talking of a derivative relation. It is possible that no account of this relation will be forthcoming until a satisfactory theory of mental representations is developed. It is possible that such a theory of mental representations would render the derivative theory otiose. It is even possible that language cannot be usefully related to the world without an intervening mental representation. Is Fodor after all a reconstructed Behaviorist?

As Steve Isard remarks in his unpublished reply, it fair takes the breath away to have one of the authors of "The structure of a semantic theory" coolly toss off "if what you mean by a semantic theory is an account of the relation between language and the world..." as if no other possibility had ever entered his head. The reader may recall that in that paper, Katz and Fodor (1963) argued that the only possible treatment of the effects of linguistic context on the interpretation of a sentence is one in which "discourse is treated as a single sentence in isolation by regarding sentence boundaries as sentential connectives". They made no suggestions as to how such a theory would work other than claiming that the great majority of sentence breaks could be treated as *and*-conjunctions. I mention this old argument, which perhaps Fodor no longer subscribes to, simply to try to rebut his charge that procedural semantics has had a negligible pay-off even as a theory of comprehension. Proceduralists have in fact shown how linguistic context affects the construction and interpretation of referring expressions, how general knowledge can be used to disambiguate a sentence in context, and how the choice of such matters as tense and connectives is affected by contextual considerations (see Johnson-Laird, 1977a, for the references). They have made progress in an area that Fodor once wrote off as impossible. A disinterested reader may yet regard this pay-off as negligible, but how can Fodor?

Conclusion

I have shown that Fodor's critique is essentially an argument against a position that is not held. There remains only a single mystery: from what did he derive his misguided notion of procedural semantics? It is tempting to reply with a Johnsonian: "Ignorance, ma'am, pure ignorance". But, after an exercise of some scholarship, I have located the source. It is in a work from which I now quote liberally:

> The only psychological models of cognitive processes that seem even remotely plausible represent such processes as computational.
>
> But, I think, nevertheless, that the core of the empiricist theory of perception is inevitable. In particular, the following claims about the psychology of perception seem to be almost certainly true and entirely in the spirit of empiricist theorizing:
> (1) Perception typically involves hypothesis formation and confirmation.
> (2) The sensory data which confirm a given perceptual hypothesis are typically internally represented in a vocabulary that is impoverished compared to the vocabulary in which the hypotheses themselves are couched.
>
> ...what happens when a person understands a sentence must be a translation process basically analogous to what happens when a machine 'understands' (viz., compiles) a sentence in a programming language. I shall try to show ... that there are broadly empirical grounds for taking this sort of model seriously.

> It may be that complex concepts (like, say, 'airplane') decompose into simpler concepts (like 'flying machine'). We shall see ... that this sort of view is quite fashionable in current semantic theories; indeed, some or other version of it has been around at least since Locke. But it may be true for all that, and if it is true it may help.
>
> ...a compiler which associates each formula in the input language with some formula in the computing language can usefully be thought of as providing a semantic theory for the input language... On the present account then, it would be plausible to think of a theory of meaning for a *natural* language (like English) as a function which carries English sentences into their representations in the putative internal code.

There they all are – the computational metaphor, the compiler as semantic theory, the sensation language of empiricism – all the notions that Fodor castigates unsparingly. And who is this benighted author? Why, none other than the Procedural Grandmother of them all, Jerry Fodor (1976). Fodor refutes himself, not procedural semantics.

References

Davidson, D. (1967) Truth and meaning. *Synthese, 17,* 304–323.

Davies, D. J. M. and Isard, S. D. (1972) Utterances as programs. In D. Michie (ed.), *Machine Intelligence, 7,* Edinburgh University Press, Edinburgh.

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

Fodor, J. A. (1976) *The Language of Thought,* Harvester, Sussex.

Fodor, J. A. (1978) Tom Swift and his Procedural Grandmother. *Cog., 6,* 229–247.

Johnson-Laird, P. N. (1977a) Procedural semantics. *Cog., 5,* 189–214.

Johnson-Laird, P. N. (1977b) Psycholinguistics without linguistics. In N. S. Sutherland (ed.), *Tutorial Essays in Psychology, Vol. I,* Erlbaum, Hillsdale, N.J.

Johnson-Laird, P. N. and Quinn, J. G. (1976) To define true meaning. *Nature, 264,* 635–636.

Longuet-Higgins, H. C. (1972) The algorithmic description of natural language. *Proc. Roy. Soc. Lond. B., 182,* 255–276.

Marr, D. and Nishihara, H. K. (1976) Representation and recognition of the spatial organization of three-dimensional shapes. MIT AI Laboratory Memorandum 377, Cambridge, Mass.

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

Ogden, C. K. (1930) Basic English. Kegan Paul, Trench, and Trubner, London.

Putnam, H. (1975) Philosophy and our mental life. *Philosophical Papers, Vol. 2: Mind, Language and Reality,* Cambridge University Press, Cambridge.

Robbin, J. W. (1969) *Mathematical Logic: A First Course,* Benjamin, New York.

Schank, R. C. (1972) Conceptual dependency: a theory of natural language understanding. *Cog. Psychol., 3,* 552–631.

Scott, D., and Strachey, C (1971) Toward a mathematical semantics for computer languages. *Proceedings of the Symposium on Computers and Automata,* Polytechnic Institute of Brooklyn, April, 1971.

Waltz, D. L. (1975) Understanding line drawings of scenes with shadows. In P. H. Winston (ed.), *The Psychology of Computer Vision,* McGraw-Hill, New York.

Winograd, T. (1971) Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. MIT, AI Laboratory, Report A.I.-TR-17., Cambridge, Mass.

Winograd, T. (1974) Five lectures on artificial intelligence. Artificial Intelligence Laboratory, Memo AIM No. 240, Stanford University, California.

Woods, W. A. (1973) Meaning and machines. *Proceedings of the Int. Conf. on Computational Linguistics,* Pisa, Italy.

Woods, W. A. (1975) What's in a link: foundations for semantic networks. In D. G. Bobrow and A. Collins (eds.), *Representation and Understanding: Studies in Cognitive Science.* Academic Press, New York.

Woods, W. A. (1978) Procedural semantics as a theory of meaning. Paper delivered at the Sloan Workshop on Computational Aspects of Linguistic Structure and Discourse Setting, May, 1978, University of Pennsylvania.