Cognition, 5 (1977) 189 - 214

Procedural Semantics*

PHILIP N. JOHNSON-LAIRD
University of Sussex

Abstract

The aim of this paper is to present an outline of a theory of semantics based on the analogy between natural and computer programming languages. A unified model of the comprehension and production of sentences is described in order to illustrate the central "compile and execute" metaphor underlying procedural semantics. The role of general knowledge within the lexicon, and the mechanism mediating selectional restrictions, are re-analyzed in the light of the procedural theory.

"Procedural semantics" is an expression that gained currency first in the discussion of computer programming languages like Fortran and Algol. 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. Their semantics consists of the procedures that the computer is instructed to execute. If, for example, a programming language permits an instruction like: x and $y \Rightarrow$, it might mean that the computer is to add the values of x and y, and to print the result.

There are usually two steps involved in running a program of instructions written in some high-level programming language. The first step is to compile the program, which consists of translating it into the operational code of the particular machine to be used. Compilation depends heavily on the syntax of

*The ideas in this paper were developed in collaboration with George A. Miller to whom I owe a very considerable intellectual debt. Professor Miller also delivered an earlier version of the paper to the Conference on Philosophy and Psychology at Cornell University, 2 April, 1976. Tom Bever and Jacques Mehler were kind enough to suggest that the paper be put into a form suitable for publication. I am indebted to Stephen Isard and Mark Steedman for many helpful comments, ideas, and criticisms, though of course they should not be blamed for my mistakes. The earlier version of the paper was also revised in the light of *some* of Jerry Fodor's pungent remarks in his unpublished critique, "Tom Swift and His Procedural Grandmother".
The research was in part supported by a grant from the Social Science Research Council (Great Britain) to the Laboratory of Experimental Psychology, Sussex University.

the programming language; getting all the commas and parentheses right is a notoriously important and often frustrating task for people just learning to write programs. The output of the compilation will be a compiled program, coded in a language that the machine can recognize and execute. The second step is to take this compiled program, along with the data it is to operate on, and to run it. If all goes well, the output is the result that the programmer wanted to obtain. How a programmer determines that the output really is what he wanted to compute is the problem of procedural semantics. For simple procedures that are well understood, like the operations of arithmetic or Boolean algebra, the semantics is relatively straightforward; a programmer can determine whether his program means what he wanted it to mean by simply checking that the machine gives the same results that he obtains when he does a few sample computations himself. In large and elaborate programs, however, it is often difficult to determine whether the result of the programmed computation is actually what the programmer intended to compute.

How to prove that a program does what the programmer claims it does is a difficult question and nothing more will be said about it here. The purpose of mentioning it at all is simply to indicate what "procedural semantics" means in this domain of discourse: procedural semantics deals with the meaning of procedures that computers are told to execute.

This discussion moves a step closer to our present concerns when we consider how computers might be used to perform various operations as a consequence of natural language inputs – messages coded, not in Fortran or Algol, but in English or Russian. It is natural to carry over the "compile and execute" strategy to natural language processing. For example, if a program is to accept questions and to provide answers written in English, the first step would be to translate the English question into a program for computing an answer or for finding the answer in some prestored data base. That is to say, the first step is to compile the question, which entails treating English questions in the same way Fortran or Algol statements would be treated. The second step is to run the compiled program in order to obtain the information required for an answer, and the final step is to formulate it in an English phrase or sentence that the machine can type out.

Obviously, natural languages like English were not designed as programming languages; to treat them as such requires considerable ingenuity on the part of a programmer. The compilation must be guided by natural language syntax, which is considerably more complicated than the syntax of artificial programming languages, and an important component of the system must be a parsing mechanism sensitive to the grammatical structure of sentences. This part of the system might be called procedural syntax, and it has been conceived in many different ways. There are parsers that try to build up analyses from the smallest constituents to the largest ones. There are parsers that operate more predictively and look at each point in a sentence for the sort of constituents they expect to find (as we shall illustrate below). Some parsers build up a chart of the different possible alternative syntactic analyses (Kay, 1975), whereas others check the semantic coherence of a proposed analysis before accepting it (Winograd, 1972). There are parsers that pursue syntactic and semantic analyses in parallel and exchange information between them either very freely in a heterarchical fashion (Woods and Makhoul, 1973) or through an extremely restricted communication channel (Reddy *et al.*, 1973). There are even systems that attempt to go straight to a semantic interpretation and then check it against the syntax of the sentence (Schank, 1972). Such a wealth of possibilities should suffice to indicate that, in the realm of natural language processing, "procedural syntax" and "procedural semantics" have been given a variety of different interpretations.

Psychologists are interested not in how a computer might compile and execute sentences, but in how people process them – in the cognitive operations that are performed in producing and understanding utterances. It is sometimes claimed that the work on the computer processing of sentences is unlikely to contribute to the scientific understanding of language because it lacks a concern for the *principles* underlying its organization and learning. There is some truth in this charge, but it may be mistaken to argue that technology cannot contribute to science or that a study is worthless because it does not elucidate some problem assumed to be central to an area. In fact, computer studies can be a useful source of hypotheses about human linguistic processing and have played a crucial part in discovering the ubiquitous role of inference in comprehension. There are a wide variety of computer systems, but they all pay allegiance in their different ways to the "compile and execute" strategy. The application of this approach to human sentence processing was first clearly formulated by Davies and Isard (1972), who pointed out that compiling and executing correspond rather naturally to stages in a person's comprehension of an utterance.

The first step that a person must perform is to compile the sentence – to translate it into a program in his or her internal mental language. This is generally an automatic and involuntary process for anyone who knows the language well. If someone says, *Pass the salt* or *When did you last see your father?* or *I'm leaving for Chicago tonight*, the listener usually compiles a corresponding program. Of course, an assertion may be treated merely as data rather than an actual program – computer programs often require data to operate on. Indeed, an assertion may not be compiled into an executable program at all (see Miller and Johnson-Laird, 1976, Sec. 3.5.7). However, it is convenient to treat the sentence as compiled into a program if it is to be verified, or if information from it is to be added to memory. The nature of the program will naturally depend on the evaluation of the sentence by higher-order procedures. Once a program is compiled, the question arises as to whether the listener should run it. Should he pass the salt? Should he tell the speaker when he last saw his father? Should he add to his store of knowledge about the speaker that he is leaving for Chicago? Choosing whether or not to execute a program is a complicated business, usually under voluntary control and often dependent on complex cognitive skills.

The advantages of this approach for the purpose of psychology are reasonably obvious. Psychologists are interested in how language is used to communicate: to make statements, to ask questions and to answer them, to make requests, and even to express invocations and imprecations. It is a virtue of the procedural approach that it places these diverse speech acts on an equal footing and provides a theoretical language for formulating hypotheses about the mental processes involved.

In order to provide a glimpse of procedural semantics in action it will be necessary to explore some of these hypotheses in detail, but first we will make a brief digression to consider the relation of this approach to the more logical approach to semantics that begins with Frege (1892) and extends down to the present day in the work of many philosophers and logicians. At first glance the two approaches may seem totally unrelated, but further consideration reveals some interesting similarities. Take, for example, the logical distinction between extensions and intensions. In model-theoretic semantics an intension is a function from possible worlds to truth values, and an extension is the truth value for a particular world. In procedural semantics there is a similar distinction between a procedure and the result of executing it. Thus 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 that the program returns when it has been executed.

This parallel also serves to highlight an advantage of the procedural approach for the purposes of psychological theory. Bertrand Russell once remarked that the essential business of language is to assert or deny facts about the world, a claim that reflects a logician's emphasis on truth and falsity. This bias is certainly not without its philosophical critics; it may simply be a historical accident arising from the fact that the first great achievement in model theory was Tarski's (1936) recursive definition of truth. But whereas model-theoretic semantics regards the extension of a sentence as its truth value, procedural semantics admits a much wider range of extensions. A truth value is but one of the possible results of executing a program; others include answers to questions, compliance with requests, additions to knowledge, modification of plans, and so on. These various consequences of language use are all of interest to psychologists, who gain little insight into their problems from a semantic theory that contemplates nothing but the truth value of sentences that appear magically out of a social vacuum.

A related aspect of the logical approach to language is also worthy of comment. The tradition has been to treat semantics as a branch of mathematics, and the analysis of natural language as involving elaborate meta-mathematical concepts. A psychologist must respect the rigor that such an approach provides, but it does have the effect of removing semantics beyond the reach of empirical test. Since the psychologist is interested in the particular system of cognitive operations that people employ, and since it is an empirical task to discover what that system is, model-theoretic approaches offer only indirect solutions to his problem.

Perhaps the chief advantage of a procedural approach is that the "compile and execute" strategy forces the theorist to consider processes as well as structures. We can illustrate this point by considering a particular procedural model that was developed as a result of a growing dissatisfaction with theories based on linguistic theory.

According to transformational grammar, the surface structure of a sentence is formally derived from an underlying level of representation, or "deep structure", in which its fundamental grammatical relations are made explicit. The derivation is by way of grammatical transformations that permute constituents, delete them, and so on. However, grammatical transformations do not correspond to psychological processes, and it turns out that there is little or no unequivocal evidence that a representation of deep structure plays any part in understanding or speaking a sentence. Effects attributed to its role in comprehension may equally well be attributed to meaning (see Johnson-Laird, 1970; Fodor, Bever, and Garrett, 1974, p. 270). Its status in speaking is even more problematical because sentences are likely to be spoken in the order in which they are planned, whereas the order of constituents in deep structure is for many sentences markedly different to their surface order. Moreover, as Fodor *et al.* (1974, pp. 393 - 7) point out, it is not obvious what psychological processes could lead from meaning to deep structure, or from deep structure to surface structure. It might be argued that the concept of deep structure should accordingly be abandoned by psycholinguists. In our view, the real problem is to reconcile its linguistic necessity with the exigencies of human information processing, and a way to do so has gradually emerged in the development of a procedural theory of comprehension (Miller and Johnson-Laird, 1976; Johnson-Laird, 1977).

We will describe a model of part of this theory that was implemented in a computer program by Mark Steedman – a process that led to the introduction of some new ideas (see Steedman and Johnson-Laird, 1977). The program answers questions about a simple one-dimensional universe of discourse that consists in 'particles' moving to and fro, colliding with each other, and so on. It comprehends and produces sentences using a novel device, a semantic transition network (STN), which, while sensitive to deep structure relations, does not set up an explicit representation of them: in understanding a sentence, it goes directly from the sentence to its meaning, and in producing a sentence, it goes directly from meaning to the sentence. The STN is a familiar parsing device, a recursive augmented transition network, modified so that it builds up, not a syntactic representation of a sentence, but its semantic representation. Once this modification is made, it is a simple matter to use the device to produce sentences.

A simple augmented transition network for parsing declarative sentences is illustrated in Figures 1 and 2 (see Thorne, Bratley, and Dewar, 1968;

Figure 1. Sentence network:

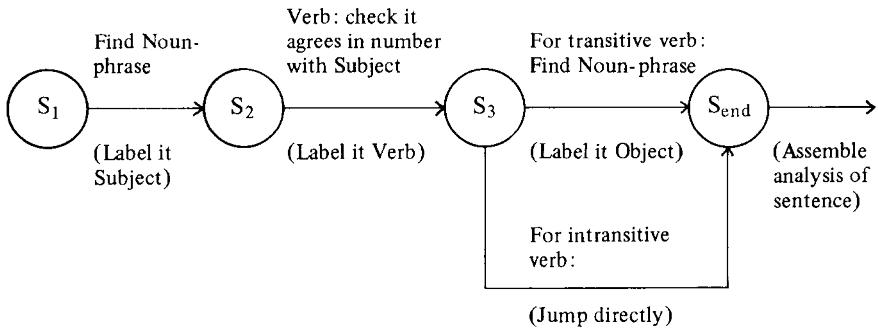
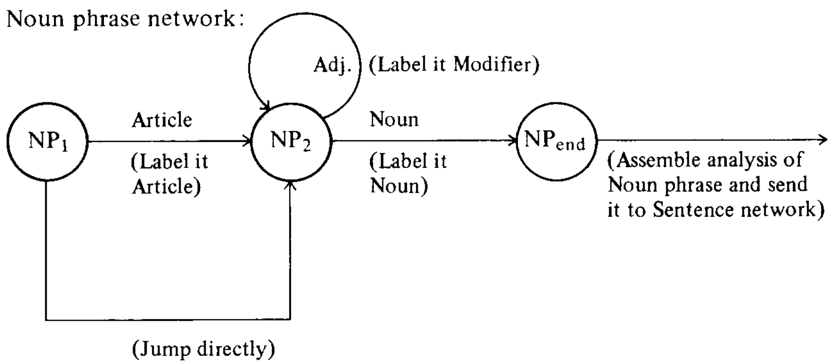


Figure 2. Noun phrase network:



Woods, 1970; Wanner and Maratsos, 1975, for further accounts from the standpoints of linguistics, artificial intelligence, and psycholinguistics, respectively). The network in Figure 1 is for parsing declarative sentences, and the network in Figure 2 is a special subcomponent for parsing the noun phrases within them. The numbers in the nodes play no part in the actual process, but are merely simple mnemonics. The system works by making transitions from node to node: a transition can be made only if the item currently being parsed satisfies the test specified above the arc. When a transition is made, the action specified beneath the arc is carried out. It will be noted that some arcs have no test associated with them – their action is carried out without parsing a word, and some arcs have only the action of jumping directly to the next node. A computer program that implements such a system needs to keep track of where in the network it is currently operating in order to ensure that it makes appropriate jumps from one component to another. It also needs a lexicon in which the syntactic categories of words are identified. If the system is unable to make any further transition but has not come to the end of a sentence, then it halts: the sentence is, as far as it is concerned, an ungrammatical one. Of course, some blockages may arise simply as a result of taking the wrong arc out of a node from which there are several. A variety of strategies can be implemented to try alternative routes through the network. The way in which the augmented transition network parses a sentence is illustrated in Table 1. Although in this case the output of the device is a sort of surface structure decorated with terms denoting deep structure relations, an augmented transition network can equally well be devized to build up a more orthodox representation of deep structure.

The STN builds up a direct *semantic* interpretation of a sentence. In order to describe its operation, we must first give an account of the way in which information about the universe of discourse is represented in the model. A typical brief history of the universe is represented by the following set of assertions in a data-base:

Object Y is at location A at time 0:	[Y AT A O]
Object X is at location B at time 0:	[X AT B O]
Object Z is at location C at time 0:	[Z AT C O]
Event E1 consists in Y moving from location A to location B from time 1 to time 2:	[E1 Y MOVE FROM A TO B START 1 FINISH 2]
Event E2 consists in X moving from location B to location C from time 2 to time 3:	[E2 X MOVE FROM B TO C START 2 FINISH 3]

Table 1. *An illustrative example of a parsing carried out by the augmented transition network depicted in Figures 1 and 2.*

Input sentence: "Leslie brought a cake"

Active node	*Test*	*Action carried out*
SI	Find a Noun phrase	Jump to Noun phrase network.
NP_1	Article	*Leslie* fails test; try another arc.
NP_1	–	Jump directly to next node.
NP_2	Adjective	*Leslie* fails test; try another arc.
NP_2	Noun	Label *Leslie* as a Noun, i.e., Noun: *Leslie*.
NP_{end}	–	(NP: Noun: *Leslie*). Jump to Sentence network.
S_1	–	Subject = (NP: Noun: *Leslie*).
S_2	Verb	Verb = *brought*
S_3	Find a Noun phrase	Jump to Noun phrase network.
NP_1	Article	Article: *a*
NP_2	Adjective	*Cake* fails test; try another arc.
NP_2	Noun	Noun: *cake*.
NP_{end}	–	(NP: Article: *a* Noun: *cake*). Jump to sentence network.
S_3	–	Object = (NP: Article: *a* Noun: *cake*)
S_{end}	–	(Sentence Subject = (NP: Article: *a* Noun: *cake*) Verb = *brought* Object = (NP: Noun: Leslie)

Event E3 consists in Z moving from location C to location D from time 3 to time 4:	[E3 Z MOVE FROM C TO D START 3 FINISH 4]

This data-base is implemented using the higher-order programming language, PICO-PLANNER (Anderson, 1972). PLANNER languages are extremely useful for manipulating sets of facts and for drawing conclusions from them (see Hewitt, 1971; Winograd, 1972). Their most important feature for our purposes is that they allow a goal to be specified to find some fact in the data-base, which they then seek to satisfy either directly by finding the fact or else indirectly by deducing it from other facts in the data-base. They are accordingly indifferent as to whether information is specified in terms of semantic primitives or in terms of more complex elements of meaning. This feature reflects our intuition that the decomposition of the meanings of words into their primitive semantic constituents is not an invariable prerequisite for the comprehension of sentences.

Figure 3. *The syntactic tests used by the STN in the program.*

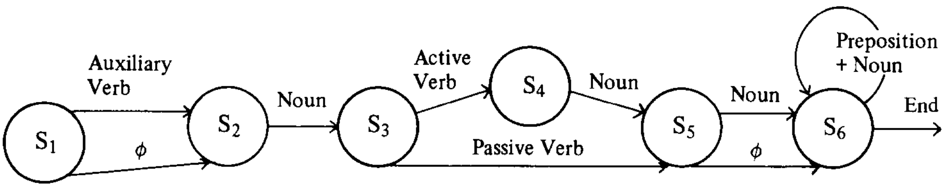


Figure 3 shows the extremely simple syntactic tests used by the STN implemented in the program. As the STN analyses a sentence, it compiles a series of semantic goals, and each goal is immediately executed with respect to the information about the history of the universe. An example of the process is summarized in Table 2. Here all the goals are satisfied, and since the program is restricted to yes/no questions, it responds, "yes". If one of the goals should fail, the program constructs a 'helpful' answer. There are two such sorts of answer, depending on the nature of the goal that fails.

Table 2. *An illustrative example of the comprehension of a question by the STN depicted in Figure 3*

Input question: "Did X hit Z at (location) C?"

Active node	*Test*	*Action carried out*	
S_1	Auxiliary verb	Jump to next node.	
S_2	Noun	Store *X* for use in constructing semantic goal.	
S_3	Active verb	Compile semantic goal:	(EVENT HIT SUBJ X)
		Execution of goal yields the assertion:	[E2 HIT SUBJ *X*]
S_4	Noun	Compile semantic goal:	(EVENT HIT OBJ Z)
		Execution of goal yields:	[E2 HIT OBJ *Z*]
S_5	Noun	*At* fails test; try another arc.	
S_5	–	Jump to next node.	
S_6	Preposition	Store *at* for use in constructing semantic goal	
S_6	Noun	*C* is stored as location name in lexicon.	
		Compile semantic goal:	(EVENT HIT AT C)
		Execution of goal yields:	[E2 HIT AT *C*]
		All goals satisfied: respond, "Yes".	

A failure may occur in a goal that corresponds to "given" information in the question, that is to say, the questioner has taken something for granted that is in fact false. For example, the question:

Did Y hit Z at C?

takes for granted that Y hit Z, and would give rise to the answer:

No, Y did not hit Z, Y hit X.

The information for this answer is obtained by noting that a goal corresponding to a "given" constituent has failed and then determining what entity Y actually hit. A failure may also occur in that part of the question that corresponds to "new" information. For example, the question:

Was Z hit by X at A?

would give rise to the answer:

No, Z was hit by X at C.

The information underlying this answer is obtained by generating a new goal to find the location of the event. In reality, the division of sentences into "given" and "new" information is a complicated business involving surface structure, intonation contour, and context. The program takes a very simple view of the matter: the distinction is recognized primarily to illustrate the feasibility of a procedural approach to it, and to explore the use of an STN in producing sentences of different sorts.

The conventional theory of the production of sentences suggests that a speaker decides on what he wants to say, and then decides on how and in what order he wants to say it. This view is certainly implicit in the notion of mapping meaning onto deep structure, and then deep structure onto surface structure. Obviously, a speaker can make a cold-blooded decision about the order in which he wishes to express his thoughts, but such decisions seem to be relatively rare. An alternative hypothesis is that the order in which the content of a message is constructed determines the actual surface order of the sentence. This is the principle by which the program operates: it is able to do so only because an explicit representation of deep structure plays no part in its operation. The STN constructs its answer from the set of assertions that are produced in evaluating the question. The essential modification required to make it produce sentences is to swop round the tests and actions on each arc so that it now tests for certain sorts of semantic content and acts by producing words in an appropriate order. The process is illustrated in Table 3.

Table 3. *An illustrative example of the production of a 'helpful' answer by the STN depicted in Figure 3.*

The evaluation of the goals set up by the input question, "Was Z hit by X at A?" gave rise to the following series of assertions:

[E2 HIT OBJ *Z*]
[E2 HIT SUBJ *X*]
[E2 HIT AT *C*], as a result of an initial failure.

These are the input to the STN, the initial failure having triggered the response, "No".

Active Node	*Test*	*Action carried out*	
S_1	Auxiliary verb	Test fails; try another arc.	
S_1	–	Jump to next node.	
S_2	[EVENT VERB. . .VAR]	Print value of variable, VAR:	*Z*
S_3	[EVENT VERB SUBJ VAR]	Test fails; try another arc.	
S_3	[EVENT VERB OBJ VAR]	Print passive VERB:	*was hit*
S_5	[EVENT VERB SUBJ VAR]	Print *by* + VAR:	*by X*
S_6	[EVENT PREP VAR]	Print PREP + VAR	*at C*
	The complete answer is: *No, Z was hit by X at C.*		

The appeal of a theory to psychologists is likely to depend on its predictive power. We shall mention just one such aspect of the present model. The STN's use of semantic cues allows its syntactic component to be considerably simplified. A conventional transformational grammar distinguishes the underlying structure of such sentences as:

The car was pushed by the police station

and:

The car was pushed by the driver.

The first example involves a locative adverbial, whereas the second example in its more salient interpretation involves a passive *by*-phrase. As Figure 3 shows, the syntactic component of the STN analyses:

X was pushed by B

where "B" denotes a location, in an identical fashion to:

X was pushed by Y

where "Y" denotes an entity. The two sentences are distinguished in setting up their semantic goals by taking into account knowledge about "B" and "Y". In essence, the program utilizes "selectional restrictions" to aid its interpretation of sentences: it appreciates that a location such as B cannot denote the subject of a pushing, whereas an entity such as Y can play this role. This conception of a simplified syntactic process is compatible with some recently discovered facts about grammatical transformations within the cycle (e.g. passive, dative movement). Such transformations are structure-preserving, i.e., they do not move constituents to positions that cannot be generated by the rules specifying deep structures (Emonds, 1976). Thus, for example, the passive *by*-phrase is not a novel structure: it is also specified by the deep structure rules generating locative and other adverbials. The cyclical transformations are also lexically dependent, i.e., their applicability depends on the presence of the appropriate lexical items. Thus, for example, the passive can be applied when the main verb is *pay*, but it cannot be applied when the main verb is *cost*. As a result of such observations, Bresnan (1976) has argued that cyclical transformations should be replaced by lexical redundancy rules – a proposal which, as she acknowledges, is particularly compatible with an ATN parsing model. In fact, it would seem to be still more compatible with an STN designed to use its semantic knowledge in analyzing structures with more than one syntactic role. The use of semantic cues in this way is contrary to the hypothesis that syntactic and semantic processes are autonomous and do not interact (Forster and Olbrei, 1973; Forster, 1976; Garrett, 1976). Recent experimental evidence suggests that syntactic and semantic variables do interact in the way predicted by the model (Steedman and Johnson-Laird, 1977).

Anyone who has written a computer program knows that there are many occasions when it cannot initially be compiled. Programmers, of course, make mistakes in syntax. Speakers, too, make grammatical mistakes, yet listeners are generally able to understand what they are saying. This observation can only be explained on the assumption that the natural-language compiler is an extremely resourceful device – indeed, much more so than an STN, which is at best a simplified, but perhaps instructive, model of only a small part of linguistic performance.

After this glimpse of a specific model, let us turn to some more general issues which are treated in other procedural theories. One such crucial phenomenon is that the same sentence can express different propositions on different occasions of use. The procedural semanticist accordingly recognizes that the same sentence can be compiled into different programs depending on the linguistic and situational contexts in which it is uttered. Text and context often modulate the meaning of sentences in similar ways, but we will consider them separately.

First, the circumstances of an utterance. There are, indeed, numerous linguistic expressions whose interpretation depends on a knowledge of when the sentence they occur in was uttered, the participants in the discourse, and what was going on in the real world. Philosophers call such expressions indexical; linguists call them deictic. Among the more obvious deictic elements are tense, personal pronouns, and certain locative expressions that depend on a speaker's point of view. The truth or falsity of a sentence such as *You are standing in front of a rock* obviously depends on the time at which it is uttered, to whom it is addressed, and the relative positions of speaker, addressee, and rock at the time of reference. Even the interpretation of nouns, verbs, and other major categories of words can be influenced by circumstance. What you understand by the assertion *I've added a new element to the group* may well depend on whether the speaker was John Lennon or John von Neumann.

Next, consider linguistic context. It is bound to create problems for any theory that does not look beyond the bounds of sentences. Here, for example, is a simple narrative:

> When John went to New York, he saw the Empire State Building. He saw Central Park and he visited the Guggenheim Museum.

The point to bear in mind is the role of tense. The clause *he saw the Empire State Building* is in the past tense. One function of the past tense is to indicate that the time of an event is in the past with respect to the time of utterance, but not just at any such time: it demands a definite reference time with which it can identify the time of the event. In the example, the reference time is given by the subordinate clause: it is *when John went to New York.* A whole chain of sentences can be tied to a given reference time, and contextual effects can accordingly spread like a benign infection from one sentence to the next, and so on. However, a quarantine zone can be established in various ways, as Longuet-Higgins (1972) has pointed out: by changing the tense of the verb or by any other device that introduces a new reference time. For example, if the narrative continues: *When he goes to New York next time,* or *He will go to New York in October,* or *He has fond memories of the city,* then the listener grasps at once that the reference time has changed. Notice that exactly the same sorts of contextual effects are created when the discourse is hypothetical: *If John went to New York, he saw the Empire State Building. He saw Central Park and he visited the Guggenheim Museum.* Or when the discourse is plainly counterfactual: *If John had gone to New York, he would have seen the Empire State Building. He would have seen Central Park and he would have visited the Guggenheim Museum.* In both these examples each subsequent clause refers back to the hypothetical reference time established in the first clause.

We have described contextual effects as a benign infection. This, of course, is only partly true. An inappropriate reliance on context can have a disastrous effect. You may recollect the White Rabbit's evidence in *Alice in Wonderland* with its complete failure to establish antecedents for its pronouns:

They told me you had been to her,
And mentioned me to him. . .
I gave her one, they gave him two,
You gave us three or more;
They all returned from him to you,
Though they were mine before

. . . . and so on, quite incomprehensibly. Yet, strangely, sometimes when the most irreparable damage seems to have been done, with whole appendages of a sentence dropping off, context can restore the missing information. For example, an utterance such as:

Did Fred?

seems rather odd in isolation. But provided it is preceded by an appropriate context, the missing appendage may be readily regenerated:

Did Charlie pass the exam?
Yes.
Did Fred?

Such are some of the problems of deixis and context. What are their solutions?

A solution sometimes suggested is to replace any element that depends on deixis or context with an equivalent element that does not. This stratagem turns out to be extraordinarily difficult to accomplish except in the case of such eternal verities as *Two plus two equals four* where it is not necessary anyway. It typically takes a sentence like *He lived there at that time* and yields such sentences as: *Christopher Marlowe, the English poet and dramatist, resides in the Italian Town of Padova on October 15th 1582,* and so on to any degree of detail that might be considered necessary to isolate the sentence from the perils of context and deixis. This example fails at once, however, because the date *October 15th 1582* does not refer to a unique day in recorded history. At that time both the Julian and the Gregorian calendars were in operation in different places and were ten days out of phase. Thus, even the interpretation of dates can be deictic. If there is ever any form of inter-stellar communication, this problem will become particularly troublesome.

Model-theoretic semantics, whatever its virtues for handling points of reference and circumstances of use, is not the most natural way for a psychologist to pursue the epidemiology of contextual effects. A more plausible method is to keep a record of reference time, participants in the discourse, recently mentioned individuals and events, and so forth, and to ensure that this record is kept up to date. Theories of this sort, couched in procedural terms, have been advocated by various theorists (see, for example, Longuet-Higgins, 1972; Isard, 1975).

We have mentioned that the compiler must have access to a lexicon, but we have yet to consider what information the lexicon should contain, or how it should be organized. A proponent of model-theoretic semantics has argued that "we should not expect a semantic theory to furnish an account of how any two expressions belonging to the same syntactic category differ in meaning" (Thomason, 1974). This statement may be an exaggeration, but it is unlikely that a model theoretician will be concerned with the difference in meaning between, say, *table* and *chair*, or between *move* and *push*. Such matters are very much the business of psychologists, and we have recently advanced some detailed arguments in favour of a procedural approach to them (Miller and Johnson-Laird, 1976). We took the view that a lexical concept interrelates a word, rules governing its syntactic behaviour, and a schema. A schema is made up from both functional and perceptual information, and may well include information that has no direct perceptual consequences. Moreover, lexical concepts are interrelated to one another. They are organized into semantic fields that have a *conceptual core* which reflects a deeper conceptualization of the world and integrates the different concepts with the semantic field. One purpose of such an organization is to create a taxonomy that enables entities within the field to be correctly categorized and readily named.

Consider how we might represent the meaning of a simple word like *chair* in order to fulfill such demands. The simplest possible way would be to include a list of exemplars of all possible chairs, or rather to specify a predicate that would do so. A logician postulates the existence of such predicates by *fiat*; a psychologist must specify the details of a procedure that will square with what is known about the processes of perceptual identification. It is feasible that one recognizes a chair of a familiar shape by matching a mental representation of a three-dimensional prototype against the visual image, perhaps in the manner described by Marr and Nishihara (1976). However, "chairness" is not a simple property. Some chairs are more prototypical than others; a simplistic hypothesis would fail to do justice to the patent absurdity of some of Carelman's (1969) more exotic creations. A procedural definition of chair, then, is likely to involve a moderately complicated schema, which will include information about the function of chairs, if only to make it possible to identify novel designs outside the perceptual repertoire of the observer. One can easily imagine some such routine that would test whether the entity in question is *a stable manmade object having as a proper part a surface intended to support someone sitting on it and as a proper part another surface intended to provide a rest for the person's back.* Of the many problems raised by this approach to the lexicon, just two will be considered here: the problem of knowledge, and the problem of selectional restrictions. It turns out that these two problems are interrelated and both hinge on the concept of possibility (see Miller, In press).

In specifying the function of some artefact such as a chair, it is evident that one passes from the world of what is actual to the world of what is possible. An object serves the function of a chair because, among other things, it is possible to sit on it and to rest against its back. How, then, are we to accommodate the concept of possibility? A model-theoretician postulates a set of possible worlds (of which the actual world is a member) and specifies an accessibility relation between them. It then becomes feasible to carry out a semantic analysis of the modal notions that occur in natural language. Thus, to borrow an example from David Lewis (1973), the counterfactual sentence, *If kangaroos had no tails, they would topple over* means roughly: in any possible world in which kangaroos have no tails, and which resembles the actual world as much as kangaroos having no tails permits it to, kangaroos topple over. The statement of such truth-conditions is an exercise very remote from the way human beings actually evaluate counterfactuals. Once again, the logician is postulating the existence of a function *by fiat*, and it is the psychologists' task to specify the details of its operation. One obvious constraint is that with the exception of a few highly restricted domains such as tic-tac-toe no human being is capable of considering, let alone evaluating, anything more than a very restricted subset of possible alternatives to a given state of affairs. One constraint on the subset to be considered is obviously an individual's general knowledge relevant to the given state of affairs. Another constraint is the nature of the heuristic that enables certain inferences to be drawn from a combination of general knowledge and specific circumstances. Such a heuristic is necessary if only because logic cannot determine which inference should be drawn, but only whether or not a given inference is valid.

What we really need is an account of the mental processes – the heuristics – that allow inferences about whether something is possible. Let us consider a specific example: suppose someone asks you, *Is it possible for you to touch me?* One part of your general knowledge that is relevant might be expressed by the generalization: if you want to touch someone, stand next to them and raise your arm in such a way that it makes contact with their body. Another relevant item of information will surely be that you are able to raise your arms in the appropriate manner. These two items of knowledge could be represented in memory in the following way:

[LIFTARM(EGO)]

which is a simple assertion that you are able to lift your arm in the appropriate way, and by the following PLANNER-like procedure:

```
(GOAL(TOUCH(X,Y))
        GOAL(NEXT(X,Y))
        AND
        GOAL(LIFTARM(X)))
```

This says roughly that if the goal is for X to touch Y, then there are two subgoals to be achieved: first, ensure that X is next to Y, and then ensure that X lifts an arm in an appropriate way. The ordering of these goals is important: it is no use raising your arm if you are not next to Y.

Let us suppose that you have appraised the situation and that you are next to the speaker. This fact will be represented by the following item in your updatable record of circumstances – we can ignore here the apparatus for dealing with time:

[NEXT(EGO,SPEAKER)]

The speaker's question was: *Is it possible for you to touch me*? which might be compiled as the following goal:

POSSIBLE(TOUCH(EGO,SPEAKER)).

When this program is executed, the instruction POSSIBLE would elicit a procedure that would try everything it could in order to derive the goal, and, in particular, it would search all the GOAL-procedures in the knowledge base for ones that match the pattern of the desired goal. In this way it would discover the procedure we defined earlier, and the values EGO and SPEAKER would be assigned to the local variables in that procedure. This would create the subgoal: FIND(NEXT(EGO,SPEAKER)). The required assertion is in the updatable record of circumstances. Similarly, the second subgoal is satisfied by the assertion in the knowledge base that you are able to lift your arms. Hence, the test is satisfied and you would be able to answer "Yes". In some situations *Is it possible for you to touch me*? would be understood, not as a yes/no question, but as an indirect request for the addressee to touch the speaker. Such subtleties can be handled expeditiously by procedural systems but to develop that aspect of the theory would require too long a digression.

There is a price to be paid for representing knowledge directly in a procedural form: the system demands a separate statement of each different inferential use of any given item of general knowledge (Winograd, 1975). For example, the same item of general knowledge that we have already utilized could also be used to infer that someone must be next to you because they just touched you. But, for this inference we would need to add a new procedure:

```
(GOAL(NEXT(X,Y))
        GOAL(TOUCH(X,Y))
        OR
        GOAL(NEAR(X,Y) AND NOT(BETWEEN(X,Z,Y)))
        OR . . .)
```

In order to avoid such redundancy, it is necessary to postulate three levels of representation. At the bottom level are items of general knowledge expressed as assertions. At the top level are powerful procedures that take assertions as arguments and construct specific procedures out of them. In this way, it would be feasible to express the generalization *if X is next to Y and X raises an arm appropriately then X touches Y* just once in the knowledge base, and top-level procedures would construct specific procedures out of it. As always, it is easy to draw up the menu, but rather difficult to concoct the recipes; yet it is comforting that this same division between assertions and procedures seems to be necessary in specifying information in the lexicon (Miller and Johnson-Laird, 1976).

Despite our ignorance of how such a system should be implemented, it is feasible to specify a general formulation of a procedural approach to possibility (see Miller, In Press):

> F is possible in the circumstances C if there is a set of procedures K corresponding to a body of organized knowledge such that, if the circumstances C are taken in conjunction with K, F is derived.

There are a number of pertinent psychological observations to be made about such a formulation. First, a clear distinction must be maintained between general knowledge and information about transient circumstances. This distinction has already been implicitly established in the decision to keep an updatable record of the circumstances of utterances. Second, a clear distinction must be made between the derivation of F and its actual execution. It is one thing to say that one can touch someone, and quite another to do so. The imperative "Touch me!" would compile the program: ACHIEVE (TOUCH(EGO, SPEAKER)). If you decide to comply, then running this program would convert subgoals into states of affairs to be achieved. Three sets of circumstances are relevant to the construction of this program: (a) if you are already touching the speaker, the request was probably not felicitous – a higher-order procedure might lead to the response *I already am touching you*; (b) if you are not touching the speaker but are next to him, the program involves nothing more than simply lifting your arm, as already described; and (c) if you are not near enough to reach him, the subgoal: ACHIEVE (NEXT(EGO,SPEAKER)) must be evaluated, and your store of knowledge would be consulted for information about procedures required to achieve this goal. If the circumstances are not suitable for a direct execution of a procedure to move closer, then a further sub-goal might be created to modify those circumstances. If this exploration of possibilities eventually leads to a compound procedure all of whose components are known to be possible, you can conclude that it is possible to comply. Let us refer to such a possible procedure as a 'plan' (Miller, Galanter, and Pribram, 1960). Whether or not you wish to comply – whether you decide to execute the plan – would depend on other circumstances extraneous to your comprehension of the command. The point, however, is that the process of determining whether or not something is possible corresponds to the development of a plan for achieving it. Conversely, if you can construct a plan for achieving something, you will believe that its achievement is possible.

This formulation is not completely adequate as a description of how *possible* and its cognates are used in ordinary speech. One can certainly imagine a situation in which a person is unable to construct an appropriate plan for achieving a particular goal – either he lacks the information or fails to make the right inference – yet nothing in his knowledge base leads him to conclude that it would be impossible. In such situations people are usually inclined to say that the goal is possible or, if they are being cautious, that it may be possible – they do not know any reason why it is not possible. Moreover, one can also imagine a situation in which it is necessary to achieve the goal, in the sense that the person cannot find any circumstances in which it could be avoided. In ordinary language it is not customary to say that something which is unavoidable is merely possible – it sounds odd to say you can obey the laws of gravity when you must obey them. These refinements do not pose real difficulties for procedural proposals, although they do lead to some complications in the formulation of *possible*: if a goal cannot be avoided under any circumstances, it is not possible, but necessary; if a goal cannot be achieved under any circumstances, it is impossible; if a goal can be achieved under some circumstances but not under others, it is possible; and if no circumstances are known under which it would be impossible, it is possibly possible.

This phrasing is deliberately intended to parallel the kind of formulation a psychologically motivated possible-worlds theorist might adopt, but with circumstances-under-which-a-person-knows-what-to-do substituted for possible worlds. The purpose here is not to refute any logical formulation, but rather to propose some plausible psychological mechanisms whereby the obvious logical relations might be realized. If one imagines that a person who has found his goal unattainable under the existing circumstances continues to search through his stored knowledge for circumstances under which it would be attainable, the output of a successful search would be a set of circumstances under which he would know how to proceed, and that output sets a new subgoal. The ontological commitment of the procedural approach is more conservative, and room is left for a person to fail to formulate a plan for achieving his goal.

This latter point is of considerable importance, since human failures in solving complex problems are notorious and must be accounted for in any plausible psychological theory. Several sources of such failures are apparent in the procedural formulation. An individual may lack the relevant general knowledge. He may misperceive or be misinformed about the actual circumstances. Most important of all, he may have the relevant information but be unable to derive a plan. The available heuristics that enable human beings to draw conclusions from premises are not such as to guarantee that derivations will always be found even when they exist. There are a number of classic experimental demonstrations of subjects' inability to grasp what is possible. Maier (1931) showed, for instance, that people are particularly inept in appreciating that an object with one well known function – a pair of pliers, say – might be used for an entirely different purpose – as the bob on a pendulum.

The evaluation of possibilities plays as important a role in procedural semantics as does the evaluation of truth. When sentences are compiled as programs, the obvious question to ask is whether those programs are executable – whether it is possible to carry them out. It is natural, therefore, that questions of possibility should arise at many points in the system. We have already indicated that the intension of the word *chair* must include a statement of function along with a description of perceptual properties; even at this level it is necessary to evaluate whether the function is possible with a candidate instance as well as whether the perceptual description is true of it. These lexical questions of possibility are not limited to nouns, of course. They also arise with many verbs.

Consider, for example, a verb like *lift* which, in one of its uses means that the logical subject of the verb does something that causes the logical object of the verb to travel upward. A causal relation between the entities denoted by the subject and object is a critical component of this meaning of *lift*, and the evaluation of such causal relations raises more questions of possibility. If what the subject did caused the object to travel upward, then it must have been the case that, if the subject had not acted as it did, the object would not have travelled upward, other things remaining the same. This latter paraphrase is, of course, a counterfactual, and counterfactuals live in the realm of possibility. If one encounters the sentence *The mouse lifted the elephant*, considerations of possibility immediately assign it to a realm of toy animals or animated cartoons.

But how do such considerations operate? Let us take the following sentence as an example: *The Smiths saw the Rocky Mountains flying to California.* The problem is to explain how one knows immediately that it is not the Rockies that were flying to California. The fact that mountains do not fly is so obvious that you may feel that there could not be any difficulty here. But if you go through the standard account of how the sentence is to be disentangled, you will see that there really is a problem.

The standard theory that we owe to Katz and Fodor (1963) would suggest that the verb *to fly* imposes a semantic restriction on its subject argument, that is to say, it takes only certain sorts of subjects. Such rules are generally called 'selectional restrictions'. A transparent example of a selectional restriction is the verb *to love*, which plainly demands that its logical subject denote, at the very least, an animate being. We want to place an analogous restriction on the underlying subject of *fly*. What should it be? We could start with a simple list: birds can fly, planes can fly, bats, wasps, bees, flies, kites, rockets, locusts, flying fish, and so on, can all fly. But then, of course, plates can fly through the air, or any other sort of object a person chooses to throw. It appears that there is a distinction to be drawn between those objects that can fly of their own volition or are self-propelled and those that are propelled by other forces. This distinction is all very well, but it does not explain how we grasp that mountains do not fly. A mountain is not a bird or a self-propelled vehicle, but how do we categorize the class of entities that can be thrown, carried, or projected through the air?

We might start marking all relevant items in the lexicon, but this strategy does not work because it depends on factual knowledge: *a plane* can fly, but *a wrecked plane* cannot. And what about people – can people fly? Well, it depends what is meant: they cannot fly by flapping their arms (outside fairy stories, that is), but of course it *was* the Smiths who were flying to California. An alternative answer, perhaps, is that we simply have as a fact in our knowledge base that mountains do not fly. The trouble with this approach is that it seems to require enormous amounts of negative information to be laid down in memory: walls don't fly, lawns don't fly, and so on. Multiply this example by all other properties and relations for all selectional restrictions and the system becomes unthinkable. Negative information is probably not stored in the lexicon except for a few obvious correctives to misconception such as that spiders are not insects, or whales are not fish.

It seems to be a mistake to concentrate on mountains, so let us reconsider flying. Its meaning can be decomposed into the following components:

FLY(X): (THROUGH(TRAVEL)) (X,AIR)

The critical concept is plainly TRAVEL, since mountains cannot swim, either. What we really need to evaluate is the expression: POSSIBLE (TRAVEL(MOUNTAINS)). Thus, once again, possibility enters the semantic system.

There are probably a number of items of general knowledge that will suffice to determine that mountains cannot fly: a mountain is a natural elevation of the earth's surface having considerable mass (American Heritage Dictionary); massive parts of the earth's surface are fixed relative to other such parts except in the case of earthquakes; if X is a fixed part of Y then X travels when Y travels, but X does not travel with respect to Y. This analysis *must* be on the right lines if only because it proves the opposite of what we want: mountains *can* fly, because they fly through space as part of the earth. The inferential system must establish that, if X is flying to California, then X is traveling with respect to a fixed part of the earth's surface; consequently, X cannot be a fixed part of the earth and therefore (barring earthquakes) X cannot be the Rocky Mountains. Notice that if the circumstances C were to involve massive earthquakes, or other similar world-shattering or mountain-unfixing events, they could modulate the evaluation of the function and allow the other interpretation of the participle.

It may not be readily apparent that this device of introducing possibility into the assignment of values to a verb's variables has a profound effect on the status of selectional restrictions. An extreme statement of this effect would be to say that selectional restrictions become totally unnecessary in a way that meets Savin's (1973) criticisms of them as theoretical entities. Savin pointed out that selectional restrictions seem to be arbitrary excresences tagged on to the semantic representations of lexical items. They are also fixed and determinate. That is to say, if they indicate that there is something anomalous about *A chair loved a table*, special measures will have to be taken to guard against the same evaluation of the perfectly sensible question: *Could a chair love a table*? Their determinate nature also insulates them from the effects of circumstance, which makes it difficult to explain why, in a context where the dish ran away with the spoon, there may be nothing anomalous about a chair falling in love with a table. The inferential mechanism that has been introduced seems to eliminate the need for selectional restrictions: their apparent effects would arise naturally from the semantic decomposition of verbs. Thus, for example, if one sense of the verb *to lift* is *to do something that causes some other thing to travel upward*, then a putative subject is tested to determine whether it could *do something*, and a putative object is tested to determine whether it could *travel upward*. Hence, the selection of arguments for a verb is a direct consequence of the components of its meaning. Likewise, the process is not fixed and determinate but dependent on the circumstances of an utterance.

The total elimination of selectional restrictions, however, is too extreme. It would be very inefficient to keep having to make the same inferences over and over again. A more sensible arrangement would be to keep a record of the classes of noun phrase that are invariably accepted as the values of a verb's arguments. In this way a child might come to learn genuine selectional restrictions and to appreciate that ordinarily, for example, *love* demands an animate subject and hence that there is something very odd about a sentence like *Sincerity loves Richard Nixon*. Thus, the present theory might well be construed not so much as replacing the conventional account of selectional restrictions as providing an explanation of how they could be learned in the first place, and how it is that people can cope when a sentence or a circumstance is sufficiently unconventional to fall outside the scope of what has been learnt. One other cautionary note should be sounded: there may well be certain restrictions on the arguments of verbs that are stylistic rather than semantic, and so could not be inferred in terms of possibilities based on general knowledge.

So much for our brief glimpse into the operation of a procedural semantics. The emphasis has been on the process of compiling because it raises the question of a procedural analysis of possibility – an approach that should provide an instructive contrast with other modes of semantic analysis. The contrast between model-theoretic semantics and procedural semantics in many ways resembles the contrast within Artificial Intelligence between those who favour a purely declarative knowledge base and those who favour a purely procedural knowledge base. The literature contains extreme examples of the declarative approach. Thus, McCarthy and Hayes (1969) argued at one time for the representation of all knowledge by a set of assertions. These assertions might be mobilized and utilized in problem-solving by some powerful, uniform, proof procedure. At the other extreme, Hewitt (cited by Winograd, 1975) has recently argued for an almost complete representation of knowledge in terms of procedures (or Actors as he now calls them). Neither of these two extremes seems to us particularly plausible as a basis for psychological theorizing. Hence, we have attempted to argue for a declarative knowledge base coupled to procedures that can convert its constituents into procedures.

One indirect consequence of such an approach is that it becomes possible to accommodate an idea that a number of theorists have begun to urge (e.g. Fodor, Fodor, and Garrett, 1975). It may be a mistake to consider that in the normal course of events the psychological representation of the meaning of a sentence exists as a single integral entity. Its integrity is threatened from a number of directions. First, many constituents of the sentence are likely to be directly mapped into pre-existing knowledge. Second, as Isard (1975) has emphasized, it may be necessary to run a program corresponding to one part of a sentence in order to compile a program for the rest of the sentence. This process could well occur in understanding a question such as: *Is that man over there the Archbishop of Canterbury*? A listener may decide not to look over there, that is, he may decide not to run the program required to find *that man over there.* But if he does go along with the speaker, then once he has identified the relevant individual he is free to compile the rest of the question and perhaps to attempt to answer it.

Finally, it should be emphasized that procedural semantics is more a methodology than a specific theory. There is considerable disagreement among its practitioners even about such fundamental issues as whether or not there are semantic primitives into which meanings of words are decomposed. Nevertheless, the procedural method seems to be particularly suitable for developing psychological theories about the meanings of words and sentences. It has two principal advantages. First, theories lying within its conceptual framework can be readily modeled in the form of computer programs: nothing quite so concentrates the mind as having to build such a model, and the process often leads to new ideas about the theory itself or how it should be tested. Second, it forces the theorist to consider processes. This is a signal virtue in comparison to model-theoretic and linguistic approaches to meaning that tend naturally to emphasize structure at the expense of process. Psychological processes take place in time, and so, too, do the operations of computers. Perhaps the metaphor can be pushed no further than that, but there does not seem to be any other equally viable alternative.

References

Anderson, D. B. (1972) Documentation for LIB PICO-PLANNER. School of Artificial Intelligence, Edinburgh University.

Bresnan, J. (1976) Towards a realistic model of transformational grammar. Paper presented at the Convocation on Communications, M.I.T.

Carelman, J. (1969) *Catalogue d'objets Introuvables.* Paris, André Balland.

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

Emonds, J. E. (1976) *A Transformational Approach to English Syntax: Root, Structure-preserving, and Local Transformations.* New York, Academic Press.

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

Fodor, J. D., Fodor, J. A., and Garrett, M. F. (1975) The psychological unreality of semantic representations. *Ling. Inq., 6,* 515–531.

Forster, K. I. (1976) The autonomy of syntactic processing. Paper presented at the Convocation on Communications, M.I.T.

Forster, K. I. and Olbrei, I. (1973) Semantic heuristics and syntactic analysis. *Cog. 2,* 319–347.

Frege, G. (1892) Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik, 100,* 25–50. Translated by M. Black as "On sense and reference" in P. Geach and M. Black. *Translations from the Philosophical Writings of Gottlob Frege.* Oxford, Blackwell, 1952.

Garrett, M. F. (1976) Word perception in sentences. Paper presented at the Convocation on Communications, M.I.T.

Hewitt, C. (1971) Description and theoretical analysis (using schemas) of PLANNER: a language for proving theorems and manipulating models in a robot. Ph.D. dissertation, M.I.T.

Isard, S. (1975) What would you have done if . . .? *J. Theoret. Ling., 1,* 233–255.

Johnson-Laird, P. N. (1970) The perception and memory of sentences. In J. Lyons (ed.), *New Horizons in Linguistics.* Harmondsworth, Middx., Penguin.

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

Katz, J. J. and Fodor, J. A. (1963) The structure of a semantic theory. *Lang., 39,* 170–210.

Kay, M. (1975) Syntactic processing and functional sentence perspective. In R. C. Schank and B. L. Nash-Webber (eds.), *Theoretical issues in natural language processing: an interdisciplinary workshop in computational linguistics, psychology, linguistics, and artificial intelligence.* Supplement, pp. 12–15, M.I.T.

Lewis, D. (1973) *Counterfactuals.* Cambridge, Mass., Harvard University Press.

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

Maier, N. R. F. (1931) Reasoning in humans. II. The solution of a problem and its appearance in consciousness. *J. Comp. Psychol., 12,* 181–194.

Marr, D. and Nishihara, H. K. (1976) Representation and recognition of the spatial organization of three-dimensional shapes. A.I. Laboratory, Memo 377. Cambridge, Mass., M.I.T.

McCarthy, J. and Hayes, P. (1969) Some philosophical problems from the standpoint of artificial intelligence. In D. Michie (ed.), *Machine Intelligence 4.* Edinburgh, Edinburgh University Press.

Miller, G. A. Practical and lexical knowledge. In Rosch, E., and Lloyd, B. (Eds.), *Human categorization.* Hillsdale, N.J.: Erlbaum. In press.

Miller, G. A., Galanter, E. and Pribram, K. (1960) *Plans and the structure of behavior.* New York, Holt, Rinehart, and Winston.

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

Reddy, D. R., Erman, L. D., Fennell, R. D. and Neely, R. B. (1973) The HEARSAY speech understanding system: an example of the recognition process. *Third International Joint Conference on Artificial Intelligence.* Stanford, Stanford Research Institute, 185–193.

Savin, H. B. (1973) Meaning and concepts: a review of Jerrold J. Katz's *Semantic Theory. Cog., 2,* 213–238.

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

Steedman, M. J. and Johnson-Laird, P. N. (1977) A programmatic theory of linguistic performance. In P. Smith and R. Campbell (eds.), *Proceedings of the Stirling Conference on Psycholinguistics.* London, Plenum. In press.

Tarski, A. (1936) Der Wahrheitsbegriff in den formalisierten Sprachen *Stud. Phil., 1,* 261–405.

Thomason, R. H. (1974) Introduction to R. H. Thomason (ed.), *Formal philosophy: selected papers of Richard Montague.* New Haven, Yale University Press.

Thorne, J., Bratley, P. and Dewar, H. (1968) The syntactic analysis of English by machine. In D. Michie (ed.), *Machine Intelligence 3.* Edinburgh, Edinburgh University Press.

Wanner, E. and Maratsos, M. (1975) An augmented transition network model of relative clause comprehension. Mimeo, Harvard University.

Winograd, T. (1972) *Understanding natural language.* New York, Academic Press.

Winograd, T. (1975) Frame representations and the declarative/procedural controversy. In D. G. Bobrow and A. Collins (eds.), *Representation and understanding: studies in cognitive science.* New York, Academic Press.

Woods, W. A. (1970) Transition network grammars for natural language analysis. *Communications of the Association for Computing Machinery, 13,* 591–606.

Woods, W. A. and Makhoul, J. (1973) Mechanical inference problems in continuous speech understanding. *Third International Joint Conference on Artificial Intelligence.* Stanford, Stanford Research Institute, 200–207.

Résumé

Le but de cet article est de présenter le schéma d'une théorie sémantique fondée sur l'analogie entre le langage naturel et le langage de programmation de l'ordinateur. On décrit un modèle unique de compréhension et de perception de phrases pour illustrer la métaphore centrale "compiler et executer" qui sous-tend les sémantiques des méthodes. On réanalyse à la lumière de la théorie des méthodes (procedural theory) le rôle de la connaissance générale interne au lexique et du mécanisme arbitrant les restrictions sélectives.