In this paper we develop the view that the specific interpretation of many ordinary English words can be fixed only within their context of use and not before. It describes an approach to the interpretation of natural-language commands in which the context of utterance is brought to bear on the interpretation of words used in the command. We also argue for a view of procedural semantics that is grounded in intentions. From this standpoint we outline the concept of a natural model for the interpretation of commands. Next we deal with those aspects of context we stress: the perceptual situation in which a command is given, the cognitive and perceptual functioning of the agent being addressed, and the immediate linguistic surround. The final section shows how our context-dependent view of the lexicon is integrated with the syntactic analysis of a sentence. Here we come into conflict with much conventional linguistic wisdom about the construction of grammars.

An important general point about much of the analysis in this paper is that it is done from the perspective of our work on instructable robots. The reasons for this emphasis are made explicit at the end of the next section.

Before turning to details we want to make some general remarks about context. The fundamental assumption of our work on the lexicon is that the precise interpretation of many words can be fixed only after the context in which the word is used has been taken into account. Contextual information, we claim, is an integral part of what a word means. Words do not in general have
determinate meaning; their semantic significance is fixed by the actual occasions of their use. Our purpose in this paper is to present evidence of the context dependency of ordinary words and to discuss the mechanisms by which the context fixes the precise interpretation of those words. We pay particular attention to the language of action—exemplified in imperatives—which places special demands on model-theoretic semantics. We argue that the language of action calls for a procedural semantics, and we discuss something of the potential of procedural semantics and the new problems it poses.

The fact that context must play a role in the interpretation of utterances has long been acknowledged. This is a familiar and obvious fact in the theory of demonstratives. A detailed logical treatment of their dependence on context is given by David Kaplan. A more general analysis of context is given by Jon Barwise and John Perry, who distinguish three aspects of context: the discourse situation (the time and place of utterance); speaker connections (a speaker's experiences, past and present, providing connections to objects, places, etc.); and resource situations (the use of one state of affairs to convey information about another). Our emphasis on the semantics of the lexicon leads in a direction that is orthogonal to the works of Kaplan and Barwise and Perry. A complete theory of context would need to include both kinds of analysis. Indeed, there is no general agreement on just what the complete role of context is. One prevailing view in both linguistics and philosophy is that an analysis of context is to be superimposed on the study of morphology, syntax, and semantics, all of which may be specified independently of the context of actual utterances. Our view is that context intrudes strongly at the lexical level and makes itself felt in ways that greatly affect traditional approaches to syntax. Suppes has examined some of these consequences for syntactic structure; we examine them again in the final section of this paper, showing the expanded force of those arguments for the language of action.

**Intentions**

There is a fundamental point about the nature of intentions that is central to our analysis. When an intention is expressed, as in “Go to the other side of the room,” the meaning of the expres-
sion does not include a detailed algorithm for executing the command. The particular path taken by the agent satisfying the command is not part of the meaning of the expressed intention. If it were, the meaning specificity of intentions and the commands expressing them would be computationally intolerable.

A vast array of scientific evidence is available to show that the central and peripheral nervous systems of humans (and other higher animals) are organized hierarchically, but also in a decentralized, pluralistic way. Instructions for detailed movements that are part of some intended action are not transmitted from the central to the peripheral nervous system—again, the computational load of doing so would be intolerable. This decentralization of responsibility for the execution of details is well reflected in our intuitive ideas about the satisfaction of a command. In no sense do we compute a specific trajectory that must be followed in crossing a room or in carrying out any other sort of movement. Even in cases of instruction, we simply cannot verbally describe a specific trajectory. Most actions in fact execute a kinematic path in the four-dimensional space-time of classical physics, the details of which are quite beyond the descriptive powers of ordinary language.

Context enters these considerations in several different ways. Part of the decentralization of action execution is that the expressed intention satisfied by the execution ignores details of context that must be taken into account at a lower level of the agent's system in order for an action to be successfully carried out; for example, an action that required a path through several traffic lights would not ordinarily have been guided by any mention of the traffic signals in the verbal expression of the intention. But the traffic lights would automatically—in ordinary cases—be taken care of at a lower level. Notice that this sketchy character of the expressed intention with respect to any actual path taken is characteristic both of intentions expressed about one's own future behavior and of commands given to another agent.

However, if an agent—person or robot—is asked to fetch a book from a table on the other side of the room, and if the agent knocks over a chair in doing so, in ordinary circumstances we regard the movement of the agent as satisfying only partially or rather poorly the request made. Expressed intentions carry with them a bundle of ceteris paribus conditions that impose a variety
of constraints on the specific procedures actually executed. These ceteris paribus conditions are not given concretely or in advance but depend on the particular context in which an action is carried out. From a psychological standpoint, these conditions become embodied in habits. Learning in all its forms is needed for the unspoken consonance between intentions and habits to develop. If someone is driving a car while thinking intently about some problem, the person will, as we say, automatically stop for red lights. Other cases of habit are even more ordinarily a matter of automatic, that is, unreflective, response. If I am walking somewhere on an errand, the motor control and perceptual feedback required for normal walking operate quite outside of consciousness. These habits are efficient but unconscious ones. Moreover, even if we try to reflect consciously on how we are walking, our cognitive insight into the details is poor unless we are specialists in the psychology and neurophysiology of such matters. And even specialists can be directly aware of only very limited aspects of their own procedures of movement.

The kinds of considerations just set forth support the view that habits take care of many contextual details of action execution—the expected details, we might want to add. Before we circle back to semantics, we will introduce a useful distinction about actions. The distinction we have in mind is familiar in the theory of events as ordinarily developed in probability theory. Suppose we roll two dice and bet on the event of a sum of eight coming up. Several different outcomes will realize this event, namely the pairs (6, 2), (5, 3), (4, 4), (3, 5), and (2, 6). In the elementary probability cases, the description of the outcomes is nearly as simple as the description of the event, but in more complicated, less regimented cases this is not so. If, for example, a meteorologist forecasts moderate rain tomorrow afternoon, he does not begin to describe the many different configurations of the atmosphere, that is, outcomes, that could produce this event. Notice that the elementary probability cases strip away by obvious convention most of the details of the outcomes. Essentially, the process of getting to an outcome is not recorded at all. Only the end result matters, provided—and this is an important proviso—the process satisfies a set of mostly unstated ceteris paribus conditions. The analysis of these conditions is no part of elementary probability theory, but is a necessary part of professional practice for those who love to shoot
craps. The most obvious standard condition is that the dice must be thrown against the vertical side of the crap table opposite the shooter before coming to rest on the horizontal surface of the table. It has been known since the time of Poincaré that the detailed analysis of the motion of dice from the instant of their being thrown until they come to rest, that is, the analysis of the process of reaching the outcome, is a matter of great mathematical difficulty. The listing of the possible outcome results for the pair of dice is trivial, but the full listing or description of the possible outcome processes is in fact impossible.

In the case of actions, the initial distinction we have in mind is between an event and an action outcome or specific action. It will also be useful occasionally to distinguish between process and result. We therefore introduce the following technical terms: “event result,” “event process,” “specific-action result,” and “specific-action process.”

We can now turn back to semantics. The semantics of commands have, from a process standpoint, an appalling lack of concreteness. When Susan says to an agent “Bring me the book on the table,” we naturally tend to think that the commands satisfaction is evaluated just in terms of the result—what we have termed the event result. Here we are close to the situation in elementary probability theory. In the pretty little model of elementary probability theory, the command to the crapshooter “Roll an eight!” has a simple result-semantics. The command is satisfied by any of the pairs listed above, and not satisfied by any of the other possible pairs. No context to worry about. No ambiguity. The same is true of the command “Bring me the book” if only the result-semantics consisting of the pair (brought the book, did not bring the book) is considered. But lurking in the background are those nebulous and troublesome ceteris paribus conditions. An ideal process-semantics of the book command should, at the specific-action process level, consist of all possible acceptable paths of movement to the table and back, together with a probabilistic measure of their likelihood of occurrence. But this is hard enough to do for the simple, idealized models of classical statistical mechanics. It is out of the question in a situation like the present one, in which the component forces determining the motion are so diverse and subtle. Something less detailed is essential; this is reflected in the inevitable vagueness of
ordinary descriptions of such processes. Ordinary language, like ordinary conscious thinking, is oriented toward results, not processes.

Even this last claim is too general. There are many devices in ordinary language for distinguishing between process and result. The many distinctions of aspect and tense that are available in English and other languages reflect important semantic features that are essential for accurate and subtle communication. It is just that we do not usually think in terms of any very elaborate schemes for expressing at the appropriate level of detail the semantics of process referred to in ordinary talk.

Although the usual discussions of aspect in English center on the indicative mood, it is easy to generate imperative examples. Contrast the perfective (1) to the imperfective (2):

(1) Stop at the table.
(2) Keep going until you reach the table.

With respect to the process-result distinctions introduced above, we immediately think of (1) in terms of results. The case of (2) is less clear. The imperfective aspect suggests process, but all the same the primary semantic evaluation would probably be result oriented, unless the agent stopped on the way to the table. We are not really certain about this, but when possible a result is probably looked for.

In many cases, however, the only semantic possibility is to make some crude appraisal of process satisfaction. Consider these imperfective imperatives expressing a demand for habitual action:

(3) Take walks every day.
(4) Keep working regularly.

We would ordinarily accept very sketchy behavior reports to judge (3) or (4) satisfied—nothing like process specificity would be asked for. To use our earlier distinctions, we would accept event-process reports at quite a high level of generality.

Given the difficulties of these semantic problems, we have adopted a standard strategy. We have retreated to a simpler framework than that of human response to a verbal command or request for action. The simpler framework is that of instructable robots, which are vastly more simple and simpleminded than
people—at least at present. Such robots accept commands in a natural language such as English and use those commands to extend their basic repertoire of actions. The kind of detailed semantic analysis required for instructable robots forces us to confront problems that may remain hidden in more abstract philosophical inquiries. Even here the semantic difficulties are daunting. Although our work in this area has been going on for some time, we are far from having anything definitive to show. Nevertheless, because we can lay out the underlying procedures in an explicit and systematic manner, our robot world, in spite of its severe limitations, provides an opportunity for a kind of detailed analysis that is not possible for human execution of commands. Many of our examples will reinforce this point, particularly those in the penultimate section, which discusses intentions and procedures for the robot that has been the focus of our recent work.

**Natural Models for the Interpretation of Commands**

In a recent phase of our work on instructable robots, we have used a robotic aid that was designed to assist the physically disabled. Earlier work made use of a robot that was taught elementary mathematics⁴ All instruction to these robots is interpreted relative to a set of models that define the agent to whom instruction is being given and the perceptual situation in which the instruction takes place. The command “Find the empty space” to the arithmetic robot of R. E. Maas and Suppes, for instance, is interpreted relative to the rows and columns of an arithmetic problem. That same command given to the mobile base of the robotic aid is interpreted relative to the configuration of objects and their parts in the room in which the instruction is taking place.

One natural semantical outcome of the viewpoint of experimental robotics is that one is not interested in the set of logically possible models satisfying a given utterance or piece of discourse. In all cases of the kind of work we are considering—and we would claim for almost all natural discourse—it is appropriate to take a subset of the set of possible models by holding rigid, at the very least, ordinary mathematics and physics, but in fact a larger body of knowledge about the real world. What this larger body of knowledge is and how it restricts the set of models may be ex-
pressed as a question about how one deals with the concept of something's being possible. We are not at all talking about the kind of possibility usually thought of in terms of logical possibility but about the ordinary notion of the possible that lies behind ordinary discourse. This notion of possibility assumes as given the kinds of fixed structures familiar in ordinary talk. Moreover, we really want to say something more radical. For the completely detailed and implemented semantic analysis needed in robotics work, we restrict even more severely the fixed set of models to ones that just encompass a particular environment, for example, the room and its physical contents in the case of the robotic aid for the physically disabled. In this set of models the frozen metaphors of abstract language so common in much ordinary talk would be ruled out. Only quite literal physical language would be understood, which means that the set of models is severely restricted to models of physical phenomena.

It is necessary, however, to include in the set of models a framework for the cognitive, perceptual, and motor functioning of the agent—person or robot—to whom the commands are addressed. This means that if we are talking about the robotic aid we are not simply restricting ourselves to the physical objects in the room but must have a way of integrating the models with the cognitive and perceptual states of the robot. The language of communication will, as we envisage it, be almost entirely physical in character. In a command like “Go to the table and pour me a glass of water,” all of the terms have a direct physical interpretation. But satisfaction of the command in a set of models requires some apparatus to express as part of the model the cognitive state and perceptual and motor activity of the robot. This point has special plausibility when one considers the verbs “remember” and “look at” in commands such as “Remember where you placed the cup” or “Look at the chair to your left.”

Reducing the possible set of models to a relatively small set helps to fix the context of an utterance. The semantic content of the lexical items is given in terms of these models, and the rules of semantic composition are expressed in terms of these models. But—and this is crucial—for many words, residual contextual factors remain and the precise interpretation of a word gets fixed only on the actual occasion of its use, and gets fixed normally, to use our earlier distinction, in terms of results, not processes. It is
the residual contextual factors that are of special interest in this paper and are the subject of the next section.

Our use of a set of models to define the context of an utterance has something in common with the "commonsense metaphysics" approach to the lexicon in which core theories are constructed about physical objects—and about time, space, material, and so on—and in which the lexical items are characterized in terms of those theories. Our work is different, however, in that we focus rather more closely on specific contexts of use, and we make provision in the lexical items for those contexts, at the time of utterance, to make their contribution.

In our work on the lexicon for the language of action, our search has been for semantic content that is specific and psychologically plausible in the context in which the language is being used. Consider the word "avoid," for instance, which in many of its ordinary uses carries the sense of evading or shunning or keeping out of the way of something. It is possible to give a general characterization of what this word means, a characterization that charts the interesting relationship between avoiding, evading, shunning, and keeping out of the way. However, such a characterization of the word is neither necessary nor sufficient in the language of action. It is not enough to enable an agent to understand in a detailed way a command such as "Avoid the chair." And the agent can understand and obey that command without ever understanding the words "shun," "evade," and "keep away from." Procedural or operational denotations are most appropriate for the language of action. Verbs such as "avoid," "look at," "put," and "remember" function semantically as operations on the natural physical models and are expressed in terms of the models that define the agent's cognitive, perceptual, and motor functioning.

The idea that a natural-language utterance may be represented semantically as a procedure performed by the language user is an old one, championed at one time by T. Winograd and supported in various ways in the work of S. D. Isard, G. A. Miller and P. N. Johnson-Laird, Suppes, and J. van Benthem. But as yet little has been done to offer a theoretically grounded view of procedural semantics for natural languages comparable to the effort for programming languages that began many years ago with J. McCarthy and R. Floyd. In principle, we would like to be able to
answer the following question of adequacy: Why these procedures and not others?

We propose the following approach to procedural semantics as a way of answering that question. In broad outline, this is what we do. We describe, intuitively and informally, a class of intentions. These are intentions we want to communicate to the agent—human or robot—through the natural-language commands whose semantics we are concerned with. We then propose a set of procedures whose satisfactory execution should demonstrate that the intentions were successfully communicated. Next, we state satisfaction conditions for these procedures, that is, conditions under which the procedures can be said to have been executed satisfactorily. Finally, we construct proofs that these satisfaction conditions can be met, proofs expressed in terms of the set of models that define the robot's cognitive, perceptual, and motor functioning in the given instructional context.

In terms of the distinction stated above in the section on intentions, we intuitively tend to define satisfaction in terms of event results, the most general of the four categories we introduced. But as we said earlier, satisfaction of a command at this level assumes a variety of unspoken ceteris paribus conditions. When we are interested in a fine-tuning of behavior, as in many instructional contexts, we want to move all the way down to analyzing satisfaction in terms of specific-action processes. Take, for instance, the two commands "Go to the table" and "Go left to the table." The first command simply expresses the result we would like to see. The second command interposes a process condition—that the table be reached by moving to the left. Yet other commands are less clear about the result to be achieved. "Go left toward the table," for instance, may be satisfied even if the table is not reached. And other commands are specific about process: "Keep moving slowly to your left until you are at the table."

It is evident that the construction of proofs of adequacy will vary considerably according to the level at which satisfaction is characterized. Here is a pair of contrasting examples to make the point. Perhaps the most familiar construction of "turtle geometry" as represented in the programming language LOGO is the drawing of a circle, for which an instruction could be "Draw a circle by repeatedly going forward one unit then left one unit." In this case it is easy to prove, for a given unit of measurement rel-
ative to the display used, how nicely the generated polygon approximates a circle. However, for the command to the robotic aid "Go to the chair," which expresses the result we would like to see, what is of interest is not the particular path taken but whether the robot reaches the chair or not. We ordinarily judge the satisfaction of the command "Go to the chair," addressed to a person or a robot, in terms of results, not in terms of process—unless something alerts us to do otherwise. If, for instance, a bystander picks up the chair and carries it over to the person, we would be obliged to consider the process whereby the person came to be by the chair.

To what extent can these questions of adequacy be posed at the lexical level? Individual lexical items are themselves often thought to denote procedures, with rules of composition stipulating how these lexical procedures are combined to form a more complex procedure for the whole command. Can we ask of each lexical procedure, Why this procedure and not some other? Consider, for example, the following procedural denotation of the verb "avoid" taken in the context of our work on instructable robots. This procedure uses information about the robot’s position and the position of the object to be avoided to generate a velocity vector away from the object. The magnitude of this vector is greatest when the robot is close to the object; it declines in proportion to the robot’s distance from the object. At a distance greater than $D$ standard units of measure from the object, the velocity vector is zero. When this procedure is being executed, the robot is never allowed to get within $d$ units of the object. But equally plausible, in the absence of further argument, is another procedure that generates a velocity vector away from the object only when the robot comes within $d$ units of the object, thereby bouncing the robot around the perimeter of the region that surrounds the object.

If, as we propose, intentions are semantically primitive, the procedures that are of primary interest are those "at the level of" the intentions. What is required of the lexical procedures is whatever allows these higher-level procedures to be executed satisfactorily and is in accordance with the class of natural models. But is that the only semantic condition that lexical entries are subject to? Certainly not, and our concern in this paper is to present an important source of semantic constraints on the lexicon, namely,
the context at the time of utterance. We emphasize that context plays a central role even at the most general level of satisfaction, that of event results.

To return to the original question of adequacy, it should be clear that the approach outlined allows the following restricted question to be answered: Why these procedures? It also implicitly shows why many other procedures would not be adequate: it would not be possible to give satisfaction conditions for them, conditions, that is, that could be proved to be met in terms of the class of models. The approach we advocate therefore allows a criterion of sufficiency, but not of necessity, to be met. This is as it should be. At the level of computational detail at which we are working—which is not the level of particular hardware and software or a particular neurophysiology but is the level of particular algorithms—we should expect to find more than one procedural account that is adequate.

What remains, of course, for procedural semantics as we conceive it is to show how the procedures associated with the intentions tie up with the English commands that express those intentions. This is where semantic grammars play their role. In a semantic grammar, rules of semantic composition are attached to the phrase-structure rules of the grammar to stipulate how the denotations of individual words are combined to produce a denotation for the whole expression. We return to this step in the final section of this paper.

**Examples of Context Fixing**

Context-dependent words are not hard to find. The examples of indexical and anaphoric pronouns and adverbs of place and time are familiar. Our claim is that these are the obvious examples of a pervasive semantic phenomenon, especially in the language of action. A first, straightforward example illustrates the role of the perceptual situation in fixing the interpretation of words. This example will come as no surprise; the words of interest in it—"left" and "right"—are generally understood to have context-dependent meaning. The example is given here, however, in preparation for the examples that follow, in which the perceptual situation is required for the interpretation of words that are not as widely recognized to be context dependent.
Consider then the commands "Turn left" and "Move to the right of the chair." There are clearly several ways to interpret a command to turn left: is it to your left (the speaker's) or to my left (the one being instructed), or am I to turn left relative to some path I have been following, as one turns left along a footpath or road? Clearly, information about the orientation of the speaker, or the listener, or any path being followed, will be required for the precise interpretation of "left" in this command. For the command "Move to the right of the chair," since many chairs are thought to have their own left and right (identified by the position of the backrest), "right" could be interpreted relative to the chair and its orientation. But if the chair has no perceptually obvious front and back, "right" must be interpreted relative to the position and orientation of the speaker or relative to the position and orientation of the agent being instructed.

Contextual information is also required to fix the interpretation of intensive adjectives such as "big" and comparatives and superlatives. Our treatment of an adjective such as "large" in the instructable robot project closely follows the analysis of Suppes and E. Macken in recognizing the existence of an underlying ordering relation on the objects referred to by the noun the adjective qualifies. The denotation of the intensive adjective "large" is thought of as a procedure that uses the underlying ordering relation of size to determine if an object, the one said to be large, stands in the appropriate size relation to some criterion object. While this ordering relation is often given by the perceptual situation, particularly for adjectives such as "red" and "large" that refer to physical attributes, there are contexts of use, even for a word such as "large," in which the ordering relation is not immediately accessible by perception—take the phrase "largest donation," for instance, as in "Our company gave the largest donation to the United Way." Our emphasis, however, has been on those uses of language for which the ordering relation is given by the perceptual context.

The criterion object for an intensive adjective is also given by the context. A simple example is provided by the adjective "large" in "large book." The criterion object will typically be different when the context is a shelf of dictionaries and when it is a shelf of poetry volumes. Perhaps the most striking example of the role of the criterion object is given by the phrases "large elephant"
and "large ant." While the ordering relations for "large elephant" and "large ant" will both use some measure such as mass or girth, the criterion objects will be quite different. Here we see that the criterion object will sometimes be set not only by extralinguistic factors such as the perceptual situation but also by the immediate linguistic surround of the word: the words "elephant" and "ant" serve to limit the range of entities that may be used as the criterion object.

Many of the words we have encountered in the instructable robot project rely on the perceptual situation for their precise interpretation. We will discuss in some detail as our next example the word "next." This word is similar to the intensive adjectives in that it relies on an underlying ordering relation. There are several other words like this: the ordinals ("first," "second," "last," and so on), the adjectives "top" and "bottom," and the adjective "previous." The intuitive idea behind the semantics of "next" can best be understood if we talk about "the next x," where x may, for example, be "chair," "table," or "wooden chair." When we say "the next x," we are referring to the first x, by some ordering relation, relative to some present reference entity. Three things have to be fixed by the context for the interpretation of "next": the ordering relation, the class of x-type entities from which one will be selected, and some encompassing class of entities that are ordered by the relation. This encompassing class must be specified because it makes perfect sense to talk about the next x even when the present reference entity is not itself an x. A clear example is given by the arithmetic robot of Maas and Suppes. For most uses of "next" in the arithmetic instruction context, the ordering relation required by "next" is given by the relation vertically below, a strict partial ordering on the perceptual objects such that each perceptual object that has a successor has a unique immediate successor by this relation, and similarly for a predecessor. That is, in the usual contexts of use for "next," the robot has been, and is expected to continue, scanning down a column. Suppose the robot is focused on the blank space at the top of the tens column of an arithmetic exercise. That blank space plays the role of the reference entity for the interpretation of "next" in "the next number." Thus, for the blank space the robot is focused on to function as the reference entity for "next number," the blank spaces and the digits (numbers) in an arithmetic exercise must all stand in the relation vertically below.
A command may explicitly fix the ordering relation required for the interpretation of "next." Consider, for instance, the command "Choose the next person in order of height," in which the ordering relation is given by the phrase "in order of height." In the absence of such explicit directions, the perceptual situation imposes its own choice of ordering relation in many cases. For instance, suppose the agent being instructed is in a room containing ten chairs arranged in a row. That very arrangement of objects will tend to establish an ordering relation for sentences in which the adjective "next" qualifies the noun "chair." If the agent were positioned alongside the second chair, facing down the row toward the third chair, and if there had been no prior discourse, the command "Go to the next chair" would probably be interpreted as a command to move to the third chair. It is clear that the appropriate ordering relation must not only be available perceptually (or by some other means such as memory), it must also be established as a focus of attention. (The robotic aid at present has no ability to adduce an ordering relation from the perceptual situation. The first time the adjective "next" is used to refer to objects of a certain type, the person instructing the robot is queried for help in fixing an ordering known to the robot. That ordering is subsequently used as the default unless explicit instruction changes it.)

Sometimes two of the three contextual factors required by "next" are set explicitly by the command. Consider again the room containing only the row of chairs, with the agent at the second chair in the row. Suppose the agent were being instructed to clean the wooden chairs by applying a furniture polish, and the row included two cane chairs, one of which was in the third position and the other in the eighth. The command "Clean the next wooden chair" would then direct the agent to the fourth chair in the row, the first wooden chair relative to the present chair. In this case, the adjective "wooden" specifies the class of wooden objects, of which one must be selected, and the noun "chair" specifies the encompassing class of chairs, both wooden and cane. Because there are no objects other than chairs in the room, the class of wooden objects is a subclass of the class of chairs. The restricting class given by the adjective will not in general be a proper subset of the encompassing class, given usually by the noun. In general, the intersection of the two classes must be found before the next $x$ can be selected.
There are many different ways a command may specify the contextual factors required by "next." Consider the command "Go to the next chair to the left." Here "to the left" specifies the ordering relation, a relation, call it $L$, which could be defined informally as follows: for all $a$ and $b$, $aLb$ if and only if $a$ is positioned to the left of $b$ and within the compass of an arc of 30 degrees radiating horizontally from $b$. Consider, however, the command "Go left to the next chair." Here "left" does not make a contribution to the interpretation of "next"; it serves rather as an adverb directly qualifying the verb, acting as an extra constraint on where to go. There are many other examples like this. On one hand, in the command "Search for the next file in alphabetic order," the ordering relation behind the use of "next" is given explicitly by the phrase "in alphabetic order." In the command "Search from $A$ to $Z$ for the next file," on the other hand, that same ordering relation defines a direction in which to search, but leaves open the question of what ordering lies behind the use of "next."

There are also occasions on which the perceptual situation does not play a role in fixing the interpretation of the word "next," as in the phrase "Susan's next book." Here the underlying ordering relation is on publication date, something that is seldom available in the immediate perceptual situation.

Our last extended example in this section draws directly on the functioning of the robotic aid that is the focus of our current work in instructable robots. Here we will examine how the agent's cognitive functioning comes into play in the interpretation of the word "avoid," as used in commands such as "Go to the door, avoiding the cat on the rug" or "While avoiding the table, move three feet left."

As part of the robotic aid's basic repertoire of perceptual and motor functions, there is a procedure that provides the core interpretation for "avoid." This procedure was described earlier. It generates a velocity vector away from the object when the robot is close to the object and prevents the robot from ever getting within $d$ units of the object. At a distance greater than $D$ units from the object, the velocity vector is zero. As described earlier, the procedure has three parameters: the first specifies the object to be avoided and the second and third the distances $d$ and $D$. The value of the first parameter is given by the interpretation of the
object noun phrase in the “avoid” command. The values of $d$ and $D$ may also be fixed explicitly by the command. “Avoid the heating element by at least one foot” sets $d$ to be one foot, for instance. In the absence of such explicit instruction, however, the context of utterance must provide values for these parameters of avoidance. While perceptual feedback may tell the robot when it is within $d$ or $D$ units of the object, it is the robot’s cognitive functioning—specifically, the robot’s knowledge of the object that is to be avoided—that will set these values appropriately. In general, different objects demand different parameters of avoidance: a fire and a delicate table lamp, for instance, require distinct $d$ and $D$ values. There are further complexities arising from objects that are part of other objects. If the object is physically part of a larger object (it is the back or the leg of a chair, for instance) and if the robot does not have the motor functioning to allow it to avoid the subpart independently of the larger object (the robot cannot, for instance, maneuver around the base or around and between the legs of the chair), the whole object must be used for the setting of the $d$ and $D$ parameters.

The significant point of this extended discussion of several examples is that for many ordinary words, there must be mechanisms for fixing the precise interpretation outside the language itself. In the final section we discuss how these mechanisms are brought into play in coordination with the grammatical analysis of a sentence. The next section examines in some detail, as required for the discussion in the final section, the procedural semantics developed for the robotic aid.

**Intentions and Procedures**

We turn now to the specifics of the mobile robotic aid operating in a room containing ordinary items of household furniture. The robotic aid consists of a six-jointed arm mounted on the front of an omnidirectional mobile base. The base is fitted with sensor-equipped bumpers. The intentions we want to communicate to this robot concerning its movement across the floor are:

- that the robot go to a given region of the room
- that the robot move in a given direction
- that the robot avoid a given region
- that the robot stay within a given region
that the robot stop doing whatever it is doing at that time
that the robot perform any specific motion at a faster than normal
speed
that the robot perform any specific motion at a slower than normal
speed
that the robot speed up
that the robot slow down
that the robot pursue two goals simultaneously (the goals are not nec-
essarily achieved simultaneously)
that the robot pursue one goal after another has been achieved
that the robot pursue a goal until a given condition is met (the pursuit
of the goal will be interrupted)
that the robot repeatedly pursue a goal until a given condition is met
that the robot pursue a goal if a certain condition is met
that the robot pursue a goal whenever a certain condition is met.

The conditions we want the robot to detect are:

that a given distance has been traversed
that a given time has elapsed
that the robot's bumpers are hit
that the robot is in a certain region.

In this discussion we are interested only in the movement of
the robot across the floor. We therefore restrict our attention to
the omnidirectional base of the robot. This mobile base can best
be described as a collection of simultaneously executing motor,
perceptual, and cognitive processes that communicate with each
other under the control of a scheduler. This scheduler has seven
modes of operation, corresponding to the following seven pro-
cedures (presented here in the notation that places the procedure
name followed by its arguments in parentheses):

(Sequence $A_1, A_2, \ldots, A_n$) Execute $A_1$, then $A_2$, and so on in sequence
to $A_n$

(Parallel $A_1, A_2, \ldots, A_n$) Start $A_1$, $A_2$, $\ldots$, $A_n$ executing simulta-
neously

(If $X A$) Execute $X$, and if $X$ returns True, execute $A$

(When $X A$) Repeatedly execute $X$ until $X$ returns True,
then execute $A$

(Whenever $X A$) Repeatedly execute $X$ until $X$ returns True,
then start $A$'s execution and begin re-
peatedly executing $X$ as before

(Do $A X$) Start the execution of $A$ and repeatedly ex-
ecute $X$ until it returns True, then inter-
rupt $A$'s operation
(Repeat A X) Repeatedly execute A and then X, until X returns True

Each of these seven procedures (also known as control structures) specifies a temporal order for the execution of its argument procedures, along with any logical connections that hold between the procedures. Each A can itself specify a control structure, or one of the robot's primitive procedures (described below). Each X specifies a test procedure that returns the value True or False. One test procedure available to the robotic aid is DistanceCovered, a procedure that takes two arguments. The first argument specifies a distance in inches; the second gives the direction along which distance is measured. The procedure returns True if the distance covered since the procedure began to be executed is greater than or equal to the distance specified, False otherwise. Another test procedure is RobotInRegion, a procedure of one argument that returns True if the robot is in the region specified by the argument, False otherwise.

The overall motion of the robot base results from the simple linear sum of motions contributed by the individual procedures operating at any time. A simple motion is expressed as a two-dimensional linear velocity plus a third component for rotation. Motions are relative to one of two coordinate systems: the first is embedded within the robot and the second is given by the room in which the robot is being instructed. The linear motion may be left or right, forward or backward in the robot coordinate system, and north or south, east or west in the room coordinate system. The rotation about the vertical axis may be clockwise or counterclockwise.

Three of the primitive procedures that contribute to the movement of the robot base are of interest to our discussion. The first, a procedure of three arguments, produces movement away from an object as described in our earlier remarks on the verb "avoid." We will return to this procedure in the next section. The two other primitive motion-procedures are Piloting and Region-Seeking. The Piloting procedure takes three arguments: the first specifies whether the movement is linear or rotational; the second specifies the direction of movement (north, for instance); and the third specifies whether the default speed of the mobile base is to be increased, decreased, or not changed at all. The call (Piloting Shift Left + ), for instance, starts a process that shifts the
robot to the left at a speed one unit greater than the default speed. Calls to the Piloting procedure such as this one are usually embedded in a Do structure, with the result that the robot stops moving to the left only when the condition specified by the Do structure becomes true. The RegionSeeking procedure takes three arguments: the first specifies a region whose nearest point the robot moves toward or away from; the second argument indicates whether that movement is toward or away; and the third argument specifies speed as for the Piloting procedure. The procedure stops executing as soon as the robot reaches the region.

While we will not offer satisfaction conditions for these procedures or proofs that the conditions can be met, we want to communicate a sense of such proofs by showing the extent to which the commands, the intentions, and the procedures fit together. Consider the following three commands.

Move toward the table.
Move three feet forward.
Continue going toward the door until you have moved forward six feet.

Each of these commands expresses a distinct intention, and consequently in our analysis, despite the fact that the verb "move" occurs in each command, distinct procedural interpretations are produced. The first command uses the RegionSeeking procedure for "move," the second the Piloting procedure, and the third the procedure DistanceCovered. The partially specified interpretations of these commands are as follows. (We use square braces for the denotations; the denotation of "forward," for instance, is shown as [forward]. The speed arguments of Piloting and RegionSeeking are omitted for simplicity.)

Move toward the table.
(Sequence (RegionSeeking [the table] Toward))

Move three feet forward.
(Do (Piloting Shift [forward]) (DistanceCovered [three feet] [forward]))

Continue going toward the door until you have moved forward six feet.
(Do [going toward the door] (DistanceCovered [six feet] [forward]))
Consider also the command "Move three feet north west." (It is convenient semantically to treat "northwest" as two separate words.) The robot's Piloting procedure knows only about the four compass directions of north, south, east, and west given by the room coordinate system. The only way to accomplish movement in the northwest direction in response to this command is to simultaneously execute (Piloting Shift [north]) and (Piloting Shift [west]). A Parallel structure is thus embedded within a Do structure with the test procedure (DistanceCovered [three feet] Trajectory). The second argument value, Trajectory, specifies that distance is to be measured along a straight line (computed off a map) from the robot's position at the time the command was given to its present position.

Move three feet north west.
(Do (Parallel (Piloting Shift [north]) (Piloting Shift [west]))
 (DistanceCovered [three feet] Trajectory))

As a final example, consider the command "Go left toward the table." In response to this command, the robot will simultaneously execute (Piloting Shift [left]) and (Regionseeking [the table] Toward) embedded within a Do structure with the test (RobotInRegion [the table]). Note that a simple Parallel structure of Piloting and Regionseeking is not adequate. With that interpretation, Regionseeking would end when the robot reached the table, but Piloting would not, and the robot would continue moving left. Note too that it makes sense to issue this command only if in moving leftward the robot would indeed reach the table.

Go left toward the table.
(Do (Parallel (Piloting Shift [left]) (Regionseeking [the table] Toward)) (RobotInRegion [the table]))

One could argue that the command "Go left toward the table" does not necessarily express the intention that the robot move all the way to the table and then stop but merely that it begin to move leftward in the direction of the table. It could be seen as semantically incomplete from the point of view of the intentions listed at the beginning of this section, since it specifies neither how far to move exactly nor for how long. However, if the robot did not stop moving when it reached the table but sailed on past it or, even worse, bumped into it, we would in ordinary circumstances
consider the command to have been poorly understood or inadequately obeyed.

This observation takes us back to our earlier remarks, in the sections on intentions and the interpretations of commands, about the distinction between process and result and about the unstated ceteris paribus conditions that often accompany verbal commands. We have already noted the extent to which process conditions are present in some commands and absent in others. Satisfaction of a command that primarily expresses a desired result is not without process constraints, however, as our examples have shown. The question how these constraints are gleaned from the context is a major challenge in our work and one we have made a small start on by recognizing the role that interaction plays. So the command "Go left toward the table," for instance, will in fact not immediately be interpreted as above, but will initiate a dialogue between the robot and the speaker to determine the speaker's intent.

We end this section with some further remarks on lexical procedures and the context. As suggested by our extended discussion of "next" and other words, the context of use should fix certain details of a lexical procedure's operation. What is in fact required for the denotation of many words is a procedural schema. In the next section we show how procedural schemata for "next" and "avoid" are used to produce a procedural interpretation for the command "Avoid the next chair." We close this section with a brief discussion of the verb "pick up" as used in a command such as "Pick up the cup."

If one particular procedure functions as the interpretation of "pick up," a procedure that stipulates exactly how a cup is picked up, the details of that procedure will be inappropriate for many commands, for there are indefinitely many ways to qualify a command and so modify the action associated with it. Consider, for instance, the commands "Pick up the cup without using the handle," "Pick up the cup by its handle," and "Pick up the cup at its rim directly across from the handle." In general, it seems as if any procedure that serves as the interpretation of a verb of action must be open to an indefinite number of modifications in its actual execution. At the same time, however, the interpretation of a command such as "Pick up the cup" is in most circumstances subject to various ceteris paribus constraints—that the vertical
The question we must face is whether or not the notion of a procedural schema allows the appropriate degree of procedural variation for verbs of action such as "pick up." One additional way of looking at this problem is to identify a default way of performing each action, recognizing that in certain contexts the default must be overridden. Under this view, the study of context would have to embrace an analysis of "normal" circumstances given by partially unstated ceteris paribus conditions in which the default holds, and other circumstances in which the default is to be overridden.

On the question how defaults are overridden, one approach is to embed procedures in a highly parallel processing environment. The special circumstance that signals the override of a default will then contribute its own procedure and the parallel execution of procedures will produce the appropriately modified action. There are cases in which this approach works and cases in which it does not. Again we draw on the robotic aid for an example. On the one hand, the procedure for "go to," as used in the commands "Go to the table" and "Go to the table without hitting the chair," produces straight-line motion of the robot toward its goal. The procedure invoked by "without hitting the chair," on the other hand, produces motion that avoids the chair. Together, these procedures have the effect of modifying the robot's direct movement toward the table, allowing it to skirt around the edge of the chair. However, not so successful a story can be told for the "pick up" command. If the default action for picking up a cup has the robot grasping the handle of the cup, and if "without using the handle" keeps the robot's gripper away from the handle, the overall effect is that the cup is not grasped at all. Much remains to be done to develop a sense of context that characterizes normal circumstances, default actions, and ways to override default actions.

Grammars and the Lexicon

The process governing the synthesis of lexical procedures to form a complex procedure for the whole sentence is as follows. A semantic tree is generated from rules of semantic composition, called semantic functions, that are attached to the phrase-structure rules of the grammar. A simple example illustrates the
main ideas. Although the example features a simple context-free grammar, the process is not restricted to such grammars. The grammar currently in use for the robotic aid in fact has its phrase-structure rules augmented with constraint equations, in the spirit of lexical-functional grammars and the unification-based formalisms of S. M. Shieber, F. C. N. Pereira, L. Karttunen, and M. Kay.\(^{13}\)

Consider the following parse tree for the imperative “Avoid the next chair.” The non-terminal labels shown are I (for imperative), VP (for verb phrase), NP (for noun phrase), V (for verb), N (for noun), DA (for definite article), and ADJ\(_{ord}\) (for ordering adjective).

This parse is produced by a context-free grammar that we extend by assigning at most one semantic function to each production rule of the grammar. The resultant grammar is called a potentially denoting grammar, following Suppes.\(^{14}\) In the grammar, we use square braces to show denotations. For instance, [NP] stands for the denotation of NP, [chair] for the denotation of “chair.” The lexical denotations are, following our general strategy, relative to the set of models that define the agent being instructed and the perceptual situation in which instruction takes place. Without going into the details of the models, let us again use for our example the mobile base of the robotic aid. Suppose it is being instructed in a room containing several chairs arranged in a row. We will describe the denotations of “avoid,” “next,” and “chair.” The definite article “the” is treated syncategorematically in this context.

Following our earlier discussion of “avoid,” suppose the denotation of “avoid” is a procedural schema with three parameters that must be set to generate a specific procedure. Let us call it Proci. The first parameter specifies the object to be avoided. The second and third are the perceptual parameters of avoidance dis-
cussed earlier: the first one establishes the minimal distance the robot must maintain from the object; the second fixes the region within which the robot’s proximity to the object must be monitored. We use the familiar lambda notation for abstraction to represent procedural schemata. The schema for “avoid” is thus shown by the expression $(\lambda x \ y \ z)\text{Proc1}(x, \ y, \ z)$.

Suppose the denotation of “next” is a procedural schema, Proc2, with three parameters corresponding to the three contextual factors identified above in the section on context fixing. That is, the first specifies the objects—chairs or tables, for instance—of which one is to be selected as the next one relative to a present reference entity. The second specifies the ordering relation that holds for the encompassing class of objects. The third specifies that encompassing class of objects.

Finally, suppose the denotation of “chair” is a simple procedure, Proc3. We will not go into the details of how the robot picks out objects in its environment. We will suppose that Proc3 returns a list of all chairs in the given perceptual environment, each chair being designated by a triple $(x, \ y, \ \theta)$ that specifies its position and orientation in a coordinate system that is fixed relative to the room.

The semantic functions below stipulate how the denotation at each node of the tree is obtained from the denotations of its daughter nodes. The extended context-free grammar for the sentence “Avoid the next chair” is as follows. Note that not all contextual parameters are set within the command itself. The notation $y_c$ indicates the parameter value $y$ given by the extralinguistic context of utterance.

<table>
<thead>
<tr>
<th>Production Rule</th>
<th>Semantic Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>I $\rightarrow$ VP</td>
<td>$[I] = [VP]$</td>
</tr>
<tr>
<td>VP $\rightarrow$ V + NP</td>
<td>$[VP] = [V][NP], \ y, \ z,$</td>
</tr>
<tr>
<td>NP $\rightarrow$ DA + ADJ$_{ord}$ + N</td>
<td>$[NP] = [ADJ_{ord}][N], \ v, \ w,$</td>
</tr>
<tr>
<td>V $\rightarrow$ avoid</td>
<td>$[V] = [\text{avoid}] = (\lambda x \ y \ z)\text{Proc1}(x, \ y, \ z)$</td>
</tr>
<tr>
<td>N $\rightarrow$ chair</td>
<td>$[N] = [\text{chair}] = \text{Proc3}$</td>
</tr>
<tr>
<td>ADJ$_{ord}$ $\rightarrow$ next</td>
<td>$[ADJ_{ord}] = [\text{next}] = (\lambda u \ v \ w)\text{Proc2}(u, \ v, \ w)$</td>
</tr>
</tbody>
</table>

The extended grammar yields the following semantic tree for “Avoid the next chair.” To the left of the colon at each node is the terminal or non-terminal label. To the right of the colon is the denotation of that label.
At the top of the tree we have a procedure specified for the command “Avoid the next chair,” stated in the imperative mood. We call this the semantic interpretation of the sentence. If the sentence were a declarative, such as “The next chair is empty,” the semantic tree for it would specify a procedure that determines the truth or falsity of the declarative. Although this example uses only one operation in its semantic functions—namely, function application—in general other operations are also used.

As the semantic functions show, not all parameter values are set within the semantic tree. In examining several examples in the section on context fixing, we saw how the precise interpretation of a word may be fixed by any of several means: the immediate linguistic surround, the perceptual situation, or the cognitive functioning of the agent. We see that same variability in this example. While the VP rule given above, for instance, is required for the sentence “Avoid the next chair,” the sentence “Avoid the next chair by six inches” would require a rule such as the following one to produce the result shown on the right. Here the phrase “by six inches” sets the parameter that determines the minimal distance the robot must maintain from the object.

Once the semantic tree has identified the parameter values still to be set, mechanisms must be invoked for obtaining those values from the perceptual situation, from the agent’s cognitive functioning, and from interaction with the user. The theory of such mechanisms is as yet not very well developed, and many would
regard their consideration as outside the proper domain of semantics. But given the drastic incompleteness of meaning if context-fixing mechanisms are not invoked in most ordinary conversation and communication, we take a contrary view. Moreover, a working understanding of these mechanisms is essential for developing adequate language capabilities in instructable robots. A central point of this paper is to show how essential they are. In other publications already mentioned we have begun a more detailed study of their nature.

Some final remarks have still to be made. The example in this section is but one of many in which the appropriate semantic interpretation of a command requires syntactic rules that are not in accord with many standard approaches to syntax. Several other examples from the instructable robot project can be found in Crangle and Suppes's report; we mention one here briefly.

Consider the command "Go three feet left." Its interpretation by the robotic aid is given by the following procedure: $(\text{Do} (\text{Piloting Shift} [\text{left}]) (\text{DistanceCovered} [\text{three feet}] [\text{left}]))$. Note that "left" makes a contribution to both the Piloting procedure and the DistanceCovered procedure. In Piloting it gives a direction of movement. In DistanceCovered it supplies a direction along which distance must be measured. (The path traced by the robot's movement may wind and turn—a point more easily seen with the command "Go three feet left while avoiding the cat"—but the intention is for the robot to move left three feet, and it is distance traversed in the leftward direction that must be measured, not distance along the path.) If "left" is to make its contribution in a straightforward way to both procedures, a relatively flat tree structure is called for in the parse of this sentence, a point we will return to in some detail in the next section. We of course want to claim that our syntactic rules are not merely the result of an inappropriate assignment of semantic content to words. To produce support for our use of what we call semantically driven grammars, we close this paper with a final section on grammars and the lexicon.

Perhaps the most unusual feature of the semantically driven grammars we have developed is the flatness of the derivation trees for sentences. We can illustrate our approach by considering a pair of simple sentences. What we prove for this pair is this. If
the denotations of the lexical items are just the natural sets they should be—"tables" denotes a set of tables, and so forth—and no sets of the hierarchical kind characteristic of Montague grammars are permitted, then the trees must be flat.¹⁶ Both philosophical and scientific intuition support this restricted view of sets. Philosophically there is a natural skepticism about sets of sets of sets, and other sets higher in the hierarchy. We never talk about them in any natural concrete way. Moreover, in the part of mathematics most powerfully adapted to quantitative science, namely, classical analysis, there is only a low-level hierarchy of numbers, vectors, and functions. It seems highly unlikely that the qualitative formulations so characteristic of natural language would have in back of them a more elaborate hierarchy than is required for classical physics. A modern structuralist point is that the mind must reflect the structure of the world—at least that part we most often encounter. Perceptual language and naive physical language seem most naturally analyzed semantically by a low-level hierarchy of sets.

The analysis we give of the following pair of sentences can be extended to more complex cases:

(1) If all tables are empty, stop!
(2) If some tables are empty, stop!

In fact, for complete simplicity, we restrict ourselves just to the antecedents of (1) and (2), that is,

(1′) All tables are empty.
(2′) Some tables are empty.

First, some concepts need to be explicitly defined, even if they are familiar. A model structure of a grammar G with terminal vocabulary \( V_T \) is a pair \((D, v)\) in which \( D \) is a non-empty set and \( v \) is a partial function from \( V_T \) to a hierarchy \( H(D) \) of sets built up from \( D \) by closure under the operations of union, intersection, and other set-theoretic operations. A model structure \((D, v)\) of a grammar G is Boolean if and only if for any string \( s \) of \( V_T^+ \) for which \( v \) is defined, \( v(s) \) is a subset of \( D \). \( V_T^+ \) is the set of all finite sequences of terminal symbols, minus the empty sequence.) A potentially denoting grammar G is Boolean if and only if for any Boolean model structure \((D, v)\) of G, every semantic function of G has as its value a subset of \( D \) whenever its arguments are subsets of \( D \).
We also need a Boolean formulation of the Frege function for the top of the tree. Using \( U \) for the universal quantifier function and \( E \) for the existential quantifier function, we have for any set \( A \):

\[
U(A) = \begin{cases} 
D & \text{if } A = D \\
\phi & \text{if } A \neq D 
\end{cases}
\]

where \( \phi \) is the empty set, and

\[
E(A) = \begin{cases} 
D & \text{if } A \neq \phi \\
\phi & \text{if } A = \phi 
\end{cases}
\]

To show how a flat Boolean grammar works, we have the following top-level rules and the associated semantic functions:

\[
\begin{align*}
S & \rightarrow UQ + NP + VP & [S] &= U(\neg[NP] \cup [VP]) \\
S & \rightarrow EQ + NP + VP & [S] &= E([NP] \cap [VP])
\end{align*}
\]

Note that the non-terminal symbols \( UQ \) and \( EQ \) for the universal and existential quantifiers have no denotation but operate as control-structure words at the top level. The non-terminal label \( S \) is for sentences.

Probably most linguists think of "all tables" and "some tables" as noun phrases, and consequently assign a complex denotation to such phrases, at least when pressed semantically. We do not deny there are good reasons for wanting "all tables" and "some tables" to be noun phrases, just as there are good reasons for wanting the semantics of simple sentences like (1') and (2') to be Boolean. There is a natural clash between grammar and semantics here. Our point is that the clash cannot be avoided.

We sketch the argument behind this claim. Rather than the rules just given for a flat grammar, let the top-level rules be:

\[
\begin{align*}
S & \rightarrow NP + VP \\
NP & \rightarrow UQ + N \\
NP & \rightarrow EQ + N \\
VP & \rightarrow \text{Copula} + \text{Adj} & [VP] &= [\text{Adj}]
\end{align*}
\]

The semantic functions for the first three rules are our object of study. Suppose there were Boolean semantic functions for these three grammatical rules. Then, without any loss of generality we may assume that \( UQ \) and \( EQ \) themselves do not denote, so we must have semantic functions \( h, f, \) and \( g \) such that:
What we can then show is that there are Boolean models of (1') and (2') such that the Boolean functional equations in terms of $h$, $f$, and $g$ cannot simultaneously have a solution.

The Boolean analysis just given generalizes to relation algebras. In particular, if lexical denotations are held down to sets of physical objects and relations among such objects, then the same argument given above forces the use of flat trees for prepositional phrases, as in "Go to all empty tables" and "Go to some empty tables." The technical details of this semantic analysis of prepositional phrases is rather lengthy, and so it is not given here. But the message for the lexicon is clear: there is an uneliminable tension between syntax and semantics.
Suppes and Crangle: Context-fixing Semantics

It is a pleasure to dedicate this paper to James Urmson, with whom we both enjoyed numerous philosophical discussions during the time he was at Stanford. We want to acknowledge the support of the Center for the Study of Language and Information at Stanford and the use of Lauri Karttunen's D-PATR grammar development system.


5. See, for example, H. T. A. Whiting, ed., Human Motor Actions: Bernstein Renssessed (Amsterdam, 1984).


15. Crangle and Suppes.


17. More explicit details are to be found in Suppes, "Variable-free Semantics with Remarks."