WILLIAM J. RAPAPORT

SYNTACTIC SEMANTICS: FOUNDATIONS OF COMPUTATIONAL NATURAL-LANGUAGE UNDERSTANDING

> Language (*la langue*) is a system all of whose terms are interdependent and where the value of one results only from the simultaneous presence of the others . . . . (de Saussure 1915, p. 159.)

1. INTRODUCTION

In this essay, I consider how it is possible to understand natural language and whether a computer could do so. Briefly, my argument will be that although a certain kind of semantic interpretation is needed for understanding natural language, it is a kind that only involves syntactic symbol manipulation of precisely the sort of which computers are capable, so that it is possible in principle for computers to understand natural language. Along the way, I shall discuss recent arguments by John R. Searle and by Fred Dretske to the effect that computers can *not* understand natural language, and I shall present a prototype natural-language-understanding system to illustrate some of my claims.[1]

2. CAN A COMPUTER UNDERSTAND NATURAL LANGUAGE?

What does it mean to say that a computer can understand natural language? To even attempt to answer this, a number of preliminary remarks and terminological decisions need to be made. For instance, by 'computer', I do not mean some currently existing one. Nor, for that matter, do I mean some ultimate future piece of hardware, for computers by themselves can do nothing: They need a program. But neither do I mean to investigate whether a program, be it currently existing or some ultimate future software, can understand natural language, for programs by themselves can do nothing.

Rather, the question is whether a computer that is running (or executing) a suitable program — a (suitable) program being executed or

James H. Fetzer (ed.), Aspects of Artificial Intelligence, 81—131.

run — can understand natural language. A program actually being executed is sometimes said to be a "process" (*cf.* Tanenbaum 1976, p. 12). Thus, one must distinguish three things: (a) the computer (i.e., the hardware; in particular, the central processing unit), (b) the program (i.e., the software), and (c) the process (i.e., the hardware running the software). A program is like the script of a play; the computer is like the actors, sets, etc.; and a process is like an actual production of the play — the play in the process of being performed.[2] Having made these distinctions, however, I will often revert to the less exact, but easier, ways of speaking ("computers understand", "the program understands").

What kind of program is "suitable" for understanding natural language? Clearly, it will be an AI program, both in the sense that it will be the product of research in artificial intelligence and in the (somewhat looser) sense that it will be an artificially intelligent program: for understanding natural language is a mark of intelligence (in the sense of AI, *not* in the sense of IQ), and such a program would exhibit this ability artificially.

But what kind of AI program? Would it be a "weak" one that understands natural language but that does so by whatever techniques are successful, be they "psychologically valid" or not? Or would it be a "strong" one that understands natural language in more or less the way we humans do?[3] ("More or less" may depend on such things as differences in material and organization between humans and these ultimate computers.) I do not think that it matters or that any of the considerations I will present depend on the strong/weak dichotomy, although I do think that it is likely that the only successful techniques will turn out to be psychologically valid, thus "strengthening" the "weak" methodology.

Another aspect of the program can be illuminated by taking up the metaphor of the play. This ultimate AI program for understanding natural language might be thought of as something like the script for a one-character play. When this "play" is "performed", the computer that plays the role of the sole "character" communicates in, say, English. But we do not want it to be only a one-way communication; it must not merely speak to us, the "audience", yet be otherwise oblivious to our existence (as in Disney-like audio-animatronic performances). That would hardly constitute natural-language understanding. More give and take is needed — more interaction: the play must be an audience-participation improvisation. So, too, must the program. I'll return to this theme later (Section 3.2.1).

I said earlier that understanding natural language is a mark of intelligence. In what sense is it such a mark? Alan M. Turing (1950) rejected the question, "Can machines think?", in favor of the more behavioristic question, "Can a machine convince a human to believe that it (the computer) is a human?"[4] To be able to do that, the computer must be able to understand natural language. So, understanding natural language is a necessary condition for passing the Turing Test, and to that extent, at least, it is a mark of intelligence.

I think, by the way, that it is also a sufficient condition. Suppose that a computer running our ultimate program understands, say, English. Therefore, it surely understands such expressions as 'to convince', 'to imitate a human', etc. Now, of course, merely understanding what these mean is not enough. The computer must be able to *do* these things — to convince someone, to imitate a human, etc. That is, it must not merely be a *cognitive* agent, but also an *acting* one. In particular, to imitate a human, it needs to be able to reason about what a(nother) cognitive agent, such as a human, believes. But that kind of reasoning is necessary for understanding natural language; in particular, it is necessary for understanding behavior explainable in terms of "nested beliefs" (such as: Jan took Smith's course because she believes that her fellow students believe that Smith is a good teacher; on the importance of such contexts, *cf.* Dennett 1983 and Rapaport 1984, 1986c). Finally, the computer must also, in some sense, *want* to convince someone by pretending to be a human; i.e., it must *want* to play Turing's Imitation Game. But this can be done by *telling* it to do so, and this, of course, should be told to it in English. So, if it understands natural language, then it ought to be able to pass the Turing Test. If so, then understanding natural language is surely a mark of intelligence.

But even if understanding natural language is only a necessary condition of intelligence, the question whether computers can understand natural language is something we should care about. For one thing, it is relevant to Searle's Chinese-Room Argument, which has repidly become a rival to the Turing Test as a touchstone for philosophical inquiries into the foundations of AI (Searle 1980). For another, it is relevant to Dretske's claims that computers can't even add (Dretske 1985). One of my main goals in this essay is to show why Searle's and Dretske's arguments fail. Finally, it is a central issue for much research in AI, computational linguistics, and cognitive science. Many researchers in these fields, including my colleagues and myself, are investigating techniques for writing computer programs that, we claim, will be able to

understand stories, narratives, discourse — in short, natural language (Shapiro and Rapaport 1986, 1987; Bruder *et al.* 1986). It would be nice to know if we can really do what we claim we are able to do!

3. WHAT DOES IT MEAN TO "UNDERSTAND NATURAL LANGUAGE"?

3.1. *Syntax Suffices*

To determine whether a computer (as understood in the light of the previous section) can understand natural language, we need to determine how it is possible for *anything* to understand natural language, and then to see if computers can satisfy those requirements.

Understanding has to do with meaning, and meaning is the province of semantics. Several recent attacks on the possibility of a computer's understanding natural language have taken the line that computers can only do syntax, not semantics, and, hence, cannot understand natural language. Briefly, my thesis in this essay is that *syntax suffices.* I shall qualify this somewhat by allowing that there will also be a certain causal link between the computer and the external world, which contributes to a *certain kind* of nonsyntactic semantics, but not the kind of semantics that is of computational interest. What kind of causal link is this? Well, obviously, if someone built the computer, there's a causal link between it and the external world. But the particular causal link that is semantically relevant is one between the external world and what I shall call the computer's "mind" — more precisely, the "mind" of the process produced by the running of the natural-language-understanding program on the computer.

Before I go into my reasons for hedging on what might seem to be the obvious importance of the causal link and what this link might be, let me say why I think I have a right to talk about a computer's "mind". Consider a system consisting of a computer, an AI program (or, what is more likely, a set of interacting AI programs), and perhaps a preexisting data base of information expressed in some "knowledge representation" language. (When such data bases are part of an AI program, they tend to be called "knowledge bases", and the preexisting, background information is called "world knowledge" — "innate ideas" would also be appropriate terminology.) The system will interact with a "user" — perhaps a human, perhaps another such system. Suppose that the

system behaves as follows: It indicates to the user that it is ready to begin. (This need not be indicated by a natural-language sentence.) The user types (or otherwise interactively inputs) a sentence in, say, English. Depending on the nature of the input, the system might modify its knowledge base in accordance with the information contained in this sentence. (If the input was merely 'hello', it might not.) It may then express to the user, in English, some appropriate proposition in its knowledge base. And so the dialogue would continue. (An actual example of such a dialogue is shown in Appendix 1.)

If such a system is going to be a good candidate for one that can understand natural language, it ought to be able at least to process virtually all of what the user tells it (or at least as much as a human would), to answer questions, and, most importantly, to ask questions. What's more, it ought to do this in a fashion that at least somewhat resembles whatever it is that *we* do when we understand natural language; that is, it should probably be doing some real, live parsing and generating, and not mere pattern-matching. Under this requirement, a "strong" system would parse and generate more or less precisely as humans do; a "weak" system would parse and generate using some other grammar.

But even this is not enough. The system must also *remember* all sorts of things. It must remember things it "knew" (i.e., had in its knowledge base) before the conversation began; it must remember things it "learns" (i.e., adds to its knowledge base) during the conversation; and it must be able to draw inferences (deductively, inductively, abductively, pragmatically, etc.) — thus modifying its knowledge base — and remember *what* it inferred as well as *that, how,* and probably even *why* it inferred it.

In short, it needs a knowledge base. This is why a program such as ELIZA (Weizenbaum 1966, 1974, 1976) — which lacks a knowledge base — does *not* understand natural language, though there are many programs described in the AI literature that have knowledge bases and do some or all of these things to varying degrees (e.g., SHRDLU (Winograd 1972) and BORIS (Lehnert *et al.* 1983), to name but two). The knowledge base, expressed in a knowledge-representation language augmented by an inferencing package, is (at least a part of) the "mind" of the system. I will discuss one such system later (the one responsible for the dialogue in Appendix 1).

So, my thesis is that (suitable) purely syntactic symbol-manipulation

of the system's knowledge base (its "mind") suffices for it to understand natural language. Although there is also a causal link between its "mind" and the external world, I do not think that this link is necessary *for understanding natural language.* I shall have more to say about this later; all I shall say now is that my reasons for taking this position are roughly methodologically solipsistic: the system has no access to these links, and a second system conversing with the first only has access to its own internal representations of the first system's links. Nevertheless, given that there are in fact such links, what might they be like? I shall have more to say about this, too, but for now let it suffice to say that they are perceptual links, primarily visual and auditory.

3.2. *The Chinese-Room Argument*

Now, Searle has argued that computers cannot understand natural language (or, hence, be intelligent, artificially or otherwise). In his Chinese-Room Argument, Searle, who knows neither written nor spoken Chinese, is imagined to be locked in a room and supplied with instructions in English that provide an algorithm for processing written Chinese. Native Chinese speakers are stationed outside the room and pass pieces of paper with questions written in Chinese characters into the room. Searle uses these symbols, otherwise meaningless to him, as input and — following only the algorithm — produces, as output, other Chinese characters, which are, in fact, answers to the question. He passes these back outside to the native speakers, who find his "answers . . . absolutely indistinguishable from those of native Chinese speakers" (Searle 1980, p. 418). The argument that this experiment is supposed to support has been expressed by Searle as follows:

> [I] still don't understand a word of Chinese and neither does any other digital computer because all the computer has is what I have: a formal program *that attaches no meaning, interpretation, or content to any of the symbols.* [Therefore,] . . . no formal program by itself is sufficient for understanding . . . . (Searle 1982, p. 5; italics added — *cf.* Section 3.5, below.)

If this Chinese-language-processing system passes the Turing Test, then — according to the Test — it does understand Chinese. And indeed it does pass the test, according to the very criteria Searle sets up. So how can Searle conclude that it doesn't understand Chinese? One

reason that he offers is that the program doesn't understand because it doesn't "know" what the words and sentences *mean*:

> The reason that no computer program can ever be a mind is simply that a computer program is only syntactical, and minds are more than syntactical. Minds are semantical, in the sense that they have more than a formal structure, they have a content. (Searle 1984, p. 31.)

That is, meaning — "semantics" — is something over and above mere symbol manipulation — "syntax". Meaning is a relation between symbols and the things in the world that the symbols are supposed to represent or be about. This "aboutness", or intentionality, is supposed to be a feature that only minds possess. So, if AI programs cannot exhibit intentionality, they cannot be said to think or understand in any way.

But there are different ways to provide the links between a program's symbols and things in the world. One way is by means of sensor and effector organs. Stuart C. Shapiro has suggested that all that is needed is a camera and a pointing finger (personal communication; *cf.* Shapiro and Rapaport 1987). If the computer running the Chinese-language program (plus image-processing and robotic-manipulation programs) can "see" and "point" to what it is talking about, then surely it has all it needs to "attach meaning" to its symbols.

Searle calls this sort of response to his argument "the Robot Reply". He objects to it on the grounds that if he, Searle, were to be processing all of this new information along with the Chinese-language program, he still would not "know what is going on", because now he would just have more symbols to manipulate: he would still have no direct access to the external world.

But there is another way to provide the link between symbols and things in the world: Even if the system has sensor and effector organs, it must still have internal representations of the external objects, and — I shall argue — it is the relations between *these* and its other symbols that constitute meaning for *it*. Searle seems to think that semantics must link the internal symbols with the outside world and that this is something that cannot be programmed. But if this is what semantics must do, it must do it for human beings, too, so we might as well wonder how the link could possibly be forged for us. Either the link between internal representations and the outside world *can* be made for both humans *and* computers, or else semantics is more usefully treated as linking

one set of internal symbolic representations with another. On this view, semantics does indeed turn out to be just more symbol manipulation.

Here is Searle's objection to the Robot Reply:

> I see no reason in principle why we couldn't give a machine the capacity to understand English or Chinese, since in an important sense our bodies with our brains are precisely such machines. But . . . we could not give such a thing to a machine . . . [whose] operation . . . is defined solely in terms of computational processes over formally defined elements. (Searle 1980, p. 422.)

'Computational processes over formally defined elements' is just a more precise phrase for symbol manipulation. The reason Searle gives for his claim that a machine that just manipulates symbols cannot understand a natural language is that "only something having the same causal powers as brains can have intentionality" (Searle 1980, p. 423). What, then, are these "causal powers"? All Searle tells us in his essay on the Chinese-Room Argument is that they are due to the (human) brain's "biological (that is, chemical and physical) structure" (Searle 1980, p. 422). But he does not specify precisely what these causal powers are. (In Rapaport 1985b and 1986b, I argue that they are not even causal.)

Thus, Searle has two main claims: A computer cannot understand natural language because (1) it is not a biological entity and (2) it is a purely syntactic entity — it can only manipulate symbols, not meanings. Elsewhere, I have argued that the biological issue is beside the point — that *any* device that "implements" (in the technical sense of the computational theory of abstract data types) an algorithm for successfully processing natural language can be said to *understand* language, no matter how the device is physically constituted (Rapaport 1985b, 1986a, 1986b). My intent here is to argue, along the lines sketched out above, that being a purely syntactic entity *is* sufficient for understanding natural language.[5]

Before doing that, I think it is worth looking at some aspects of Searle's argument that have been largely neglected, in order to help clarify the nature of a natural-language-understanding program.

3.2.1. *Natural-language generation*

The first aspect can be highlighted by returning to the metaphor of the natural-language-understanding program as a one-character, audience-participation, improvisatory play. Because it is improvisatory, the script[6] of the play cannot be fixed; it must be able to vary, depending

on the audience's input. That is, a natural-language-understanding system must be able to respond appropriately to arbitrary input (it must be "robust"). This could, perhaps, be handled by a "conditional script": if the audience says $\ulcorner\varphi_1\urcorner$, then the character should respond by saying $\ulcorner\varphi_2\urcorner$, etc. But to be truly robust, to script would need to be "productive", in roughly Chomsky's sense: that is, the character in the play must be able to understand and the produce arbitrary "new" and relevant lines. In fact, it is fairly easy to have a productive *parser* for a natural-language-understanding system. I am not claiming that the problem of natural-language *understanding* has been solved, but we seem to be on the right track with respect to parsers for natural language *processing*, and, at any rate, we know the general outlines of what a suitably robust parser should look like. What's needed, however, is *generative* productivity: the ability to *ask* new and relevant questions and to *initiate* conversation (in a non-"canned" way: ELIZA — which relies purely on pattern-matching — still doesn't qualify). To be able to generate appropriate utterances, the system must have the capability to *plan* its speech acts, and, so, a planning component must be part of a natural-language-understanding system. Such a planning component is probably also needed for parsing, in order to be able to understand *why* the speaker said what he or she did. (*Cf.* Cohen and Perrault 1979; Appelt 1982, 1985; and Wiebe and Rapaport 1986.)

To the extent that these are missing from the Chinese-Room Argument, Searle-in-the-room wouldn't seem to understand Chinese. So, let us imagine that AI researchers and computational linguists have solved this problem, and that our system is equipped with a suitably productive generation grammar. Now, these productive capabilities are tantamount to general intelligence, as I argued in Section 2. The important point, however, is that this capability is a function of what's in the system's knowledge base: what can be produced by a productive generative grammar must first be in the knowledge base. To put it somewhat mundanely, I can only speak about what I'm familiar with. (To put it more esoterically, whereof I cannot speak, thereof I must be silent.)

3.2.2. *Learning and linguistic knowledge*

A second aspect of Searle's argument that I want to look at concerns the kind of knowledge that Searle-in-the-room is alleged to have — or lack. One difference that is sometimes pointed out between machine understanding and human understanding is that everything that the

machine does is explicitly coded. This is part of what is meant when it is said that computers can only do what they are programmed to do (by someone who *is* "intelligent" or who *can* understand natural language). Furthermore, this might be interpreted to mean that the system knows everything that it is doing. But this is mistaken. It can only be said to "know" what it is "aware" of, not what is merely coded in. For instance, the knowledge bases of many AI systems distinguish between propositions that are explicitly believed by the system and those that are only implicitly believed (*cf.* Levesque 1984; Rapaport 1984, 1986c, 1987). Furthermore, the parser that transduces the user's input into the system's knowledge base, as well as the generator that transduces a proposition in the knowledge base into the system's natural-language output, need not (and arguably should not) be part of the knowledge base itself. Such "knowledge" of language would be tacit knowledge, just as Chomsky said: It is coded in and is part of the overall system, but it is not "conscious knowledge". It is no different for humans: everything we know, including our knowledge of how to understand our native natural language, must (somehow) be "coded in". In other words, human and machine understanding are *both* fully coded, but neither the human nor the machine knows everything. In the Chinese-Room Argument, the human following the Chinese-language program is in the same position as a human speaking English (only in slow motion; *cf.* Hofstadter 1980): neither has conscious knowledge of the rules of the language.

Could the machine or the human *learn* the rules, and thus gain such conscious knowledge? Or could it learn *new* rules and thus expand its natural-language understanding? Surely, yes: see the work by Jeannette Neal (Neal 1981, 1985; Neal and Shapiro 1984, 1985, and 1987).

There are other roles for learning in natural-language understanding. Many (perhaps most) conversations involve the learning of new information. And it is often the case that new words and phrases, together with their meanings, are learned both explicitly and implicitly (*cf.* Rapaport 1981, and the discussion of 'swordsman' in Section 3.5, below). In all of these cases, the learning consists, at least in part, of modifications to the system's knowledge base.

It is not clear from the rather static quality of Searle's Chinese-language program whether Searle intended it to have the capability to learn. Without it, however, the Chinese-Room Argument is weakened.

3.2.3. *The knowledge base*

It should be clear by now that a knowledge base plays a central role in natural-language understanding. Searle's original argument includes a Schank-like script as part of the input, but it is not clear whether he intended this to be a modifiable knowledge base of the sort I described as the system's "mind" or whether he intended it as the rather static structure that a script (in its early incarnation) actually is. In any case, parts of the knowledge base would probably have to be structured into, *inter alia*, such frame-like units as scripts, memory-organization packets, etc. (*Cf.* Minsky 1975, Schank 1982.) The two aspects we have just considered, and part of my argument below, imply that a modifiable knowledge base is essential to natural-language understanding. (*Cf.* n. 13.)

3.3. *Dretske's Argument*

Having set the stage, let me introduce some of my main ideas by considering Dretske's argument in 'Machines and the Mental' (1985), to the effect that an external, non-syntactic semantics is needed for natural-language understanding.

According to Dretske, machines "lack something that is essential" for being a rational agent (p. 23). That is, there is something they "can't do" (p. 23) that prevents their "membership in the society of rational agents" (p. 23). That is surely a very strong claim to make — and a very important one, if true. After all, theoretical computer science may be characterized as the study of what is effectively computable. That is, assuming Church's Thesis, it may be characterized as the study of what is expressible as a recursive function — including such theoretically uninteresting though highly practical recursive functions as payroll programs. It follows that AI can be characterized as the study of the extent to which mentality is effectively computable. So, if there is something that computers can't do, wouldn't it be something that is *not* effectively computable — wouldn't it be behavior that is nonrecursive?[7] It is reasonable to expect that it is much too early in the history of AI for such a claim as this to be proved, and, no doubt, I am interpreting Dretske's rhetoric too strongly. For a nonrecursive function is in a sense more complex than a recursive one, and Dretske's line of argument seems to be that a computer is simpler than a human (or that

computer thought is more isolated than human thought): "Why can't pure thought, the sort of thing computers purportedly have, stand to ordinary thought, the sort of thing we have, the way a solitary stroll stands to a hectic walk down a crowded street?" (p. 23). Even granting that this talk about computers is to be understood in the more precise sense of Section 1, above, the ratio

$$\frac{\text{pure thought}}{\text{computers}} = \frac{\text{ordinary thought}}{\text{humans}}$$

isn't quite right. If anything, the phrase 'pure thought' ought to be preserved for the *abstraction* that can be *implemented* in computers *or* humans (or Martians, or chimps, or . . .):

$$\frac{\text{pure thought}}{\text{implementing medium}} = \frac{\text{AI program that implements pure thought}}{\text{computer}} = \frac{\text{human mental processes (ordinary thought)}}{\text{human}}$$

(*Cf.* Rapaport 1985b, 1986b.)

What is it, then, that these "simple-minded" computers can't do? Dretske's admittedly overly strong answer is:

> They don't solve problems, play games, prove theorems, recognize patterns, let alone think, see, and remember. They don't even add and subtract. (p. 24.)

Now, one interpretation of this, consistent with holding that intelligence is nonrecursive, is that these tasks are also nonrecursive. But, clearly, they aren't (or, at least, not all of them are). A second interpretation can be based on the claim that Church's Thesis is not a *reduction* of the notion of "algorithm" to that of, say, Turing-machine program on the grounds that an algorithm is an intensional entity that contains as an essential component a description of the problem that it is designed for, whereas no such description forms part of the Turing-machine program (Goodman 1986). So, perhaps, the tasks that Dretske says computers can't do are all ones that must be described in intensional language, which computers are supposed incapable of.

These two interpretations are related. For if tasks that are essentially intensional are nonrecursive, then Church's Thesis can be understood as holding that for each member of a certain class of nonrecursive

functions (namely, the essentially intensional but effectively computable tasks), there is a corresponding recursive function that is extensionally equivalent (i.e., input—output behaviorally equivalent) to it. And Dretske's thesis can be taken to be that this equivalence is not an identity. For instance, although my calculator's input—output *behavior* is identical to my own behavior when I perform addition, *it* is not *adding*.

Here is Dretske's argument (p. 25):

(1) ". . . 7, 5 and 12 are numbers."
(2) "We add 7 and 5 to get 12 . . .".
(3) Therefore, "Addition is an operation on numbers."
(4) "At best, [the operations computers perform] are operations on . . . physical tokens that *stand for.* or are interpreted as standing for, . . . numbers."
(5) Therefore, "The operations computers perform . . . are not operations on numbers."
(6) "Therefore, computers don't add."
(7) Therefore, computers cannot add.[8]

Possibly, if *all* that computers do is manipulate uninterpreted symbols, then they do *not* add. But if the symbols are interpreted, then maybe computers *can* add. So, who would have to interpret the symbols? Us? Them? To make the case parallel, the answer, perhaps surprisingly, is: them! For who interprets the symbols when *we* add? Us. But if we can do it (which is an assumption underlying premise (2)), then why can't computers? But perhaps it is *not* I who interpret "my" symbols when I add, or you when you add. Perhaps there is a dialectical component: the only reason that *I* think that *you* can add (or *vice versa*) is that *I* interpret *your* symbol manipulations (and *vice versa*). In that case, if *I* interpret the *computer's* symbol manipulations, then — to maintain the parallelism — *I* can say that *it* adds (at least as well as you add). And, take note, in the converse case, the *computer* can judge that *I* "add".

Premise (2) and intermediate conclusion (3) are acceptable. But *how* is it that we add numbers? By manipulating physical tokens of them. That is, the abstract operation of adding *is* an operation on numbers (as (3) says), but our *human implementation* of this operation is an operation on (physical) *implementations* of numbers. (Cf. Shapiro 1977, where it is argued that addition, as humans perform it, is an operation on numerals, not numbers.) So premise (4) is also acceptable;

but — as Dretske admits — if it implies (5), then the argument "shows that we don't add either" (p. 26), surely an unacceptable result.

What the argument does illuminate is the relation of an abstraction to its implementations (Rapaport 1985b, 1986b). But, says Dretske, something is still missing:

> the machine is . . . restricted to operations on the symbols or representations themselves. It has no access . . . to the *meaning* of these symbols, to the things the representations represent, to the numbers. (p. 26)

The obvious question to ask is: How do *we* gain this essential access? And a reasonable answer is: In terms of a theory of arithmetic, say, Peano's (or that of elementary school, for that matter). But such a theory is expressed in symbols. So the *symbol* '1' means the *number* 1 for *us* because it is linked to the '1' that represents 1 in the theory. All of this is what I shall call *internal* semantics: semantics as an interconnected network of internal symbols — a "semantic network" of symbols in the "mind" or "knowledge base" of an intelligent system, artificial or otherwise. The *meaning* of '1', or of any other symbol or expression, is determined by its locus in this network (*cf.* Quine 1951; Quillian 1967, 1968) *as well as* by the way it is *used* by various processes that reason using the network. (*Cf.* the "knowledge-representation hypothesis", according to which "there is . . . presumed to be an internal process that 'runs over' or 'computes with' these representational structures" (Smith 1982, p. 33).)

There's more: *My* notion of 1 might be linked not only to my internal representation of Peano's axioms, but also to my representation of my right index finger and to representations of various experiences I had as a child (*cf.* Schank 1984, p. 68). Of course, the computer's notion of 1 won't be. But it *might* be linked to its internal representation of itself in some way[9] — the computer need not be purely a Peano mathematician. But perhaps there's too much — should such "weak" links really be part of the *meaning* of '1'? In one sense, yes; in another, no: I'll discuss several different kinds of meaning in Section 3.7.

This notion of an internal semantics determined by a semantic network and independent of links to the external world — independent, that is, of an "external" semantics — is perfectly consistent with some of Dretske's further observations, though not with his conclusions. For instance, he points out that "physical activities" such as adding "cannot acquire the relevant kind of meaning merely by *assigning* them an

interpretation, by letting them mean something *to* or *for us*" (p. 26). This kind of assignment is part of what I mean by "external semantics". He continues: "Unless the symbols being manipulated mean something *to the system manipulating them*," — this is roughly what I mean by "internal semantics" — "their meaning, whatever it is, is irrelevant to evaluating what the system is doing when it manipulated them" (pp. 26-27). After all, when *I* undergo the physical processes that constitute adding, it is not only *you* who says that I add (not only *you* who assigns these processes an interpretation for *you*), but I, too. Of course, one reason that *I* assign them an interpretation is the fact that *you* do. So, *how* do *I* assign them an interpretation? If this question can be answered, perhaps we will learn how the *computer* can assign them an interpretation — which is what Dretske (and Searle) deny can be done. One answer is by my observing that *you* assign my processes an interpretation. I say to myself, no doubt unconsciously, "I just manipulated some symbols; you called it 'adding 7 and 5'. So *that's* what 'adding' is!" But once this label is thus internalized, I no longer need the link to you. My internal semantic network resumes control, and I happily go on manipulating symbols, though now I have a few extra ones, such as the label 'adding'. After all, "How would one think of associating an idea with a verbal image if one had not first come upon (*surprenait*) this association in an act of speech (*parole*)?" (de Saussure 1915, p. 37).

Dretske expresses *part* of this idea as follows: "To understand *what* a system is doing when it manipulates symbols, it is necessary to know, not just what these symbols mean, what interpretation they have been, or can be, *assigned*," — i.e., what label *you* use — "but what they mean to the system performing the operations" (p. 27), i.e., how they fit into the system's semantic network. Dretske's way of phrasing this is not quite right, though. He says, "To understand what a system is doing . . ."; but *who* does this understanding? Us, or the system? For *me* to understand what the system is doing, I only need to know *my* assignment function, *not* the system's internal network. Unless I'm its programmer, how *could* I know it? Compare the case of a human: For me to understand what *you* are doing. I only need to know *my* assignment function. Given the privacy of (human) mental states and processes, how could I possibly know yours? On the other hand, for the *system* to understand what *it* is doing, it only needs to know its own semantic network. Granted, part of that network consists of nodes (the labels)

created in response to "outside" stimuli — from you or me. But this just makes it possible for the system and us to communicate, as well as making it likely that there will be a good match between the system's interpretation and ours. This is another reason why *learning* is so important for a natural-language-understanding program, as I suggested earlier (Section 3.2.2.). Unless the system (such as Searle-in-the-room) can learn from its interactions with the interlocutors, it won't pass the Turing Test.

Dretske's point is that the computer doesn't do what we do because it can't understand what it's doing. He tries to support this claim with an appeal to a by-now common analogy:

> Computer simulations of a hurricane do not blow trees down. Why should anyone suppose that computer simulations of problem solving must themselves solve problems? (p. 27)

But, as with most of the people who make this analogy, Dretske doesn't make it fully. I completely agree that "computer simulations of a hurricane do not blow trees down." They do, however, *simulatedly* blow down *simulated* trees (*cf.* Gleick 1985; Rapaport 1986b and forthcoming). And, surely, computer simulations of problem solving do *simulatedly* solve *simulated* problems. The natural questions are: Is *simulated* solving *real* solving? Is a *simulated* problem a *real* problem?

The answer, in both cases, is 'Yes'. The simulated problem is an *implementation* of the *abstract* problem. A problem abstractly speaking remains one in any implementation: Compare this "real" problem:

> What *number* x is such that $x + 2 = 3$?

with this "simulated" version of it:

> What *symbol* s is such that the physical process we call 'adding' applied to s and to '2' yields '3'?

Both are problems. The *simulated* solution of the *simulated* problem *really* solves it and can be used to really solve the "real" problem. To return to hurricanes and minds, the difference between a simulated hurricane and a simulated mind is that the latter does "blow down trees"! (*Cf.* Rapaport, forthcoming.)

Dretske sometimes *seems* to want too much, even though he asks almost the right question:

how does one build a system that is capable not only of performing operations on (or with) symbols, but one *to which* these symbols mean **something**, a machine that, in this sense, understands **the** meaning of the symbols it manipulates? (p. 27; italics in original, my boldface.)

A system to which the symbols mean "something": Can they mean *anything*? If so, then an internal semantics suffices. It could be based on a semantic network (as in SNePS — *cf.* Shapiro and Rapaport 1986, 1987; *cf.* Section 3.6, below) or on, say, discourse representation theory (Kamp 1984 and forthcoming, Asher 1986 and 1987). The symbols' meanings would be determined solely by their locus in the network or the discourse representation structure. But does Dretske really want a machine that understands "the" meaning of its symbols? Is there only one, preferred, meaning — an "intended interpretation"? How could there be? Any formal theory admits of an infinite number of interpretations, equivalent up to isomorphism. The "label" nodes that interface with the external wourld can be changed however one wants, but the network structure will be untouched. This is the best we can hope for.

The heart of Dretske's argument is in the following passages. My comments on them will bring together several strands of our inquiry so far. First.

if the meaning of the symbols on which a machine performs its operations is . . . wholly derived from us, . . . then there is no way the machine can acquire understanding, no way these symbols can have a meaning to *the machine itself.* (pp. 27-28)

That is, if the symbols' meanings are purely external, then they cannot have internal meanings. But this does not follow. The external-to-the-machine meanings that *we* assign to its symbols are *independent* of its own, internal, meanings. It may, indeed, have symbols whose internal meanings are causally derived from our external ones (these are the "labels" I discussed earlier; in SNePS, they are the nodes at the heads of LEX arcs — cf. Section 3.6, below, and: Shapiro 1982; Maida and Shapiro 1982; Shapiro and Rapaport 1986, 1987). But the machine begins with an internal semantic network, which may be built into it ("hardwired" or "preprogrammed", or "innate", to switch metaphors) but is, in any case, developed in the course of dialogue. So it either begins with or develops its own meanings independently of those that we assign to its symbols.

Next,

> Unless these symbols have . . . an intrinsic meaning . . . independent of **our** communicative intentions and purposes, then **this meaning** *must* be irrelevant to assessing what the machine is doing when it manipulates them. (p. 28; italics in original, boldface added.)

I find this confusing: which meaning is irrelevant? Dretske's syntax seems to require it to be the "intrinsic" meaning, but his thesis requires it to be the previous passage's "meaning derived from us" (*cf.* the earlier citation from pp. 26-27). On this reading, I can agree. But the interesting question to raise is: How independent is the intrinsic meaning? Natural-language understanding, let us remember, requires conversation, or dialogue; it is a *social* interaction. Any natural-language-understanding system must initially learn *a* meaning from its interlocutor (*cf.* de Saussure 1915, p. 37, cited above), but *its* network will rarely if ever be identical with its interlocutor's. And this is as true for an artificial natural-language-understanding system as it is for us: As I once put it, we almost always misunderstand each other (Rapaport 1981, p. 17; *cf.* Schank 1984, Ch. 3, esp. pp. 44-47).

Finally,

> The machine is processing meaningful (to us) symbols . . . but the *way* it processes them is quite independent of *what they mean* — hence, nothing *the machine* does is explicable in terms of the meaning of the symbols it manipulates . . . . (p. 28)

This is essentially Nicolas Goodman's point about Church's Thesis (discussed earlier in this section). On this view, for example, a computer running a program that *we* say is computing greatest common divisors does not "know" that that is what (we say that) it is doing; so, that's *not* what it's doing. Or, to take Dretske's example (p. 30), a robot that purportedly recognizes short circuits "really" only recognizes certain gaps; it is we who interpret a gap as a short circuit. But why not provide the computer with knowledge about greatest common divisors (so-named) and the robot with knowledge about short circuits (so-named), and link the number-crunching or gap-sensing mechanisms to this knowledge?

Observe that, in the passage just cited, the machine's symbol-processing is independent of what the symbols mean *to us*, i.e., independent of their external meaning. On Dretske's view, what the machine does is inexplicable in terms of *our* meanings. Thus, he says that

machines don't answer questions (p. 28), because, presumably, "answers questions" is *our* meaning, not *its* meaning.

But from Dretske's claim it does not follow that the symbols are meaningless or even that they differ in meaning from our interpretation. For one thing, *our* meaning *could* also be the *machine's* meaning, if its internal semantic network happens to be sufficiently like ours (just as yours ought to be sufficiently like mine). Indeed, for communication to be successful, this will have to be the case. For another, *simulated* question-answering *is* question-answering, just as with simulated problem-solving. If the abstract answer to the abstract question, "Who did the Yankees lose to on July 7?", is: the Red Sox; and if the simulated answer (e.g., a certain noun phrase) to the simulated question (e.g., a certain interrogative sentence), 'Who did the Yankees lose to on July 7?', is 'the Red Sox' (or even, perhaps, the simulated team, in some knowledge-representation system); and if *both* the computer *and* we take those symbols in the "same" sense — i.e., if they play, roughly, the same roles in our respective semantic networks — then the *simulated* answer *is* an answer (the example is from Green 1961).

How are such internal meanings developed? Here, I am happy to agree with Dretske: "In the same way . . . that nature arranged it in our case" (p. 28), namely, by correlations between internal representations (either "innate" or "learned") and external circumstances (p. 32). And, of course, such correlations are often established during *conversation.* But — contrary to Dretske (p. 32) — this can be the case for all sorts of systems, human as well as machine.

So, I agree with many of Dretske's claims but not his main conclusion. We *can* give an AI system information about what it's doing, although *its* internal interpretation of what it's doing might not be the same as ours; but, for that matter, yours need not be the same as mine, either. Taken literally, computers *don't* add if "add" means what *I* mean by it — which involves what *I* do when I add and the locus of 'add' in *my* internal semantic network. But thus understood, *you* don't add, either; only I do. This sort of solipsism is not even methodologically useful. Clearly, we want to be able to maintain that you and I both add. The reasons we are able to maintain this are that the "label" nodes of *my* semantic network match those of yours *and* that my semantic network is structurally much like yours. How much alike? Enough so that when we talk to each other, we have virtually no reason to believe that we are misunderstanding each other (*cf.* Russell 1918,

Quine 1969, Shapiro and Rapaport 1987; note, however, that in the strict sense in which only I add, and you don't, we *always* systematically misunderstand each other — *cf.* Rapaport 1981). That is, we can maintain that we both add, because we *converse* with each other, thus bringing our internal semantic networks into closer and closer "alignment" or "calibration". But this means that there is no way that we can prevent a natural-language-understanding system from joining us. In so doing, we may learn from it — and adjust to it — as much as it does from (and to) us.[10] Rather than talking about *my* adding, *your* adding, and *its* adding (and perhaps marveling at how much alike they all are), we should talk about the *abstract* process of adding that is *implemented* in each of us.

3.4. *Deixis*

My claim, then, is that an internal semantics is sufficient for natural-language understanding and that an external semantics is only needed for *mutual* understanding. I shall offer an explicit argument for the sufficiency thesis, but first I want to consider a possible objection to the effect that *deictic* expressions require an external semantics — that an internal semantics cannot handle indexicals such as 'that'.

Consider the following example, adapted from Kamp (forthcoming): How would our system be able to represent in its "mind" the proposition expressed by the sentence, "That's the man who stole my book!"? Imagine, first, that it is the system itself that utters this, having just perceived, by means of its computational-vision module, the man in question disappear around a corner. What is the meaning of 'that', if not its external referent? And, since its external referent could not be inside the system, 'that' cannot have an internal meaning. However, the output of any perceptual system must include some kind of internal symbol (perhaps a complex of symbols), which becomes linked to the semantic network (or, in Kamp's system, to the discourse representation structure) — a sort of *visual* "label". That symbol (or one linked to it by a visual analogue of the SNePS LEX arc) is the internal meaning of 'that'. (There may, of course, be other kinds of purely internal reference to the external would. I shall not discuss those here, but *cf.* Rapaport 1976, and Rapaport 1985/1986, Section 4.4.)

Imagine, now, that the sentence is uttered *to* the system, which looks up too late to see the man turn the corner. The external meaning of

'that' has not changed, but we no longer even have the visual label. Here, I submit, the system's interpretation of 'that' is as a disguised definite (or indefinite) description (much like Russell's theory of proper names), perhaps "the (or, a) man whom my interlocutor just saw". What's important in this case is that the system must interpret 'that', and whatever its interpretation is is the internal meaning of 'that'.

3.5. *Understanding and Interpretation*

This talk of interpretation is essential. I began this section by asking what "understanding natural language" means. To understand, in the sense we are discussing,[11] is, at least in part, to provide a semantic interpretation for a syntax. Given two "systems" — human or formal/artificial — we may ask, What does it mean for one system to understand the other? There are three cases to consider:

Case 1. First, what does it mean for *two humans to understand each other*? For me to understand what you say is for me to provide a semantic interpretation of the utterances you make. I treat those utterances as if they were fragments of a formal system, and I interpret them using as the domain of interpretation, let us suppose, the nodes of *my* semantic network. (And you do likewise with my utterances and your semantic network.) That is, I map your words into my concepts.

I may err: In Robertson Davies's novel, *The Manticore*, the protagonist, David Staunton, tells of when he was a child and heard his father referred to as a "swordsman". He had taken it to mean that his father was "a gallant, cavalier-like person" (Davies 1972, p. 439), whereas it in fact meant that his father was a lecher ('whoremaster' and 'amorist' are the synonyms (!) used in the book). This leads to several embarrasments that he is oblivious to, such as when he uses the word 'swordsman' to imply gallantry but his hearers interpret him to mean 'lechery'. Staunton had correctly recognized that the word was being used metaphorically, but he had the wrong metaphor. He had mapped a new word (or an old word newly used) into his concepts in the way that seemed to him to fit best, though it really belonged elsewhere in his network.

So, my mapping might not match yours. Worse, I might not be able to map one or more of your words into my concepts in any straightforward way at all, since your conceptual system (or "world view") — implemented in your semantic network — might be radically different

from mine, or you may be speaking a foreign language. This problem is relevant to many issues in translation, radical and otherwise, which I do not wish to enter into here (but *cf.* n. 13). But what I *can* do when I hear you use such a term is to fit it into my network as best I can, i.e., to devise the best theory I can to account for this fragment of your linguistic data. One way I can do this, perhaps, is by augmenting my network with a sub-network of concepts that is structurally similar to an appropriate sub-network of *yours* and that *collectively* "interprets" your term in terms of my concepts. Suppose, for example, that you are a speaker of Nuer: although your word 'kwoth' and its sub-network of concepts might not be able to be placed in 1—1 correspondence with my word 'God' and *its* sub-network of concepts (they are not exact translations of each other), I can develop my own sub-network for 'kwoth' that is linked to the rest of my semantic network and that enables me to gloss your word 'kwoth' with an account of its meaning in terms of its locus in my semantic network (*cf.* Jennings 1985). I have no doubt that something exactly like this occurs routinely when one is conversing in a foreign language.

What is crucial to notice in this case of understanding is that when I understand you by mapping your utterances into the symbols of my internal semantic network, and then manipulate these symbols, I am performing a syntactic process.

Case 2. Second, what does it mean *for a human to understand a formal language* (or formal system)? Although a philosopher's instinctive response to this might be to say that it is done by providing a semantic interpretation for the formal language, I think this is only half of the story. There are, in fact, *two* ways for me to understand a formal language. In 'Searle's Experiments with Thought' (Rapaport 1986a), I called these "semantic understanding" and "syntactic understanding". In the example I used there, a syntactic understanding of algebra might allow me to solve equations by manipulating the symbols ("move the x from the right-hand side to the left-hand side and put a minus sign in front of it"), whereas a semantic understanding of algebra might allow me to describe those manipulations in terms of a balancing-scale ("if you remove the unknown weight from the right-hand pan, you must also remove the same amount from the left-hand pan in order to keep it balanced"). Semantic understanding is, indeed, understanding via semantic interpretation. Syntactic understanding, on the other hand, is

the kind of understanding that comes from directly manipulating the symbols of the formal language according to its syntactic rules. Semantic understanding is what allows one to prove soundness and completeness theorems *about* the formal language; syntactic understanding is what allows one to prove theorems *in* the formal system.

There are two important points to notice about semantic understanding. The first is that there is no unique way to understand semantically: there are an infinite number of equally good interpretations of any formal system. Only one of these may be the "intended" intepretation, but it is not possible to uniquely identify which one. What 'adding' means to me, therefore, may be radically different from what it means to you, even if we manipulate the same symbols in the same ways (*cf.* Quine 1969, Section I, especially pp. 44-45). The second point is that an interpretation of a formal system is essentially a *simulation* of it in some *other* formal system (or, to return to talk of languages, *my* interpretation of a formal language is a mapping of its terms into my concepts), and, thus, it is just more symbol manipulation.

Syntactic understanding is also, obviously, an ability to manipulate symbols, to understand what is invariant under all the semantic interpretations. In fact, my syntactic understanding of a formal system is the closest I can get to its internal semantics, to what Dretske calls the system's "intrinsic meanings".

Case 3. Finally, what would it mean *for a formal system to understand me*? This may seem like a very strange question. After all, most formal systems just sort of sit there on paper, waiting for me to do something with them (syntactic manipulation) or to say something about them (semantic interpretation). I don't normally expect them to interpret *me*. (This is, perhaps, what underlies the humor in Woody Allen's image of Kugelmass, magically transferred into the world of a textbook of Spanish, "running for his life . . . as the word *tener* ('to have') — a large and hairy irregular verb — raced after him on its spindly legs" (Allen 1980, p. 55).)

But there are some formal systems, namely, certain computer programs, that at least have the *facility* to understand (one must be careful not to beg the question here) because they are "dynamic" — they are capable of being run. Taking up a distinction made earlier, perhaps it is the *process* — the natural-language-understanding program being run on (or, implemented by) a computer — that understands. So, what

would it mean for such a formal system to understand me? In keeping with our earlier answers to this sort of question, it would be for it to give a semantic interpretation to its input consisting of *my* syntax (my utterances considered as more or less a formal system) in terms of *its* concepts. (And, of course, we would semantically understand its natural-language output in a similar manner, as noted in Case 2.) But its concepts would be, say, the nodes of its semantic network — symbols that it manipulates, in a "purely syntactic" manner. That is, it would in fact be "a formal program that attaches . . . meaning, interpretation, or content to . . . the symbols" — precisely what Searle (1982, p. 5; cited earlier) said did not exist!

So the general answer to the general question — What does it mean for one system to understand another? — is this:

> A natural-language-understanding system S_1 understands the natural-language output of a natural-language-understanding system S_2 by building and manipulating the symbols of an internal model (an interpretation) of S_2's output considered as a formal system.

S_1's internal model would be a knowledge-representation and reasoning system that manipulates symbols. It is in this sense that syntax suffices for understanding.

The role of external semantics needs clarification. Internal and external semantics are two sides of the same coin. The *internal* semantics of S_1's linguistic expressions constitutes S_1's understanding of S_2. The *external* semantics of S_1's linguistic expressions constitutes S_2's understanding of S_1. It follows that the external semantics of S_1's linguistic expressions is the internal semantics of S_2's linguistic expressions! S_1's *internal* semantics links S_1's words with S_1's own concepts, but S_1's *external* semantics links S_1's words with the concepts of S_2.

What about "referential" semantics — the link between word and object-in-the-world? I do not see how this is relevant to S_1's *or* S_2's understanding, except in one of the following two ways. In the first of these ways, semantics is concerned with language-in-general, not a particular individual's idiolect: it is concerned with *English* — with the "socially determined" extensions of words (Putnam 1975) — not with what *I* say. This concern is legitimate, since people tend to agree pretty well on the referential meanings of their words, else communication would cease; recall the Tower of Babel. So, on this view, what does

'pen' mean? Let us say that it means the kind of object I wrote the manuscript of this essay with (I'm old-fashioned). But what does *this* mean — what does it mean to say that 'pen' means a certain kind of physical object? It means that virtually all (native) speakers of English use it in that way. That is, this view of semantics is at best parasitic on individual external semantics.

But only "at best"; things are not even that good. The second way that "referential" semantics is relevant is, in fact, at the individual level. You say 'pen'; I interpret that as "pen" in my internal semantic network. Now, what does "pen" mean for me? Internally, its meaning is given by its location in my semantic network. Referentially, I might point to a real pen. But, as we saw in our discussion of deixis, there is an internal representation of my pointing to a pen, and it is *that representation* that is linked to my semantic network, *not* the real pen. And now here is why the first view of referential semantics won't do: How does the semanticist assert that 'pen'-in-English refers to the class of pens? Ultimately, by pointing. So, at best, the semanticist can link the pen-node of some very general semantic network of English to *other* (visual) representations, and these are either the semanticist's own visual representations or else they are representations in some *other* formal language that goes proxy for the world. The semantic link between word and object is never direct, but always mediated by a representation (*cf.* Rapaport 1976, 1985a, 1985/1986). The link between that representation and the object itself (which object, since I am only a *methodological* solipsist, I shall assume exists) is a causal one. It may, as Sayre (1986) suggests, even be the ultimate source of semantic information. But it is noumenally inaccessible to an individual mind. As Jackendoff (1985, p. 24) puts it, "the semantics of natural language is more revealing of the internal representation of the world than of the external world *per se*".

Finally, some comments are in order about the different kinds of meaning that we have identified. I shall postpone this, however, till we have had a chance to look at a prototype natural-language-understanding system in operation.

3.6. *SNePS/CASSIE: A Prototype AI Natural-Language-Understanding System*

How might all this be managed in an AI natural-language-understanding

system? Here, I shall doff my philosopher's hat and don my computer scientist's hat. Rather than try to say how this can be managed by *any* natural-language-understanding system, I shall show how one such system manages it. The system I shall describe — and to which I have alluded earlier — is SNePS/CASSIE: an experiment in "building" (a model of) a mind (called 'CASSIE') using the SNePS knowledge-representation and reasoning system. SNePS, the *S*emantic *Ne*twork *P*rocessing *S*ystem (Shapiro 1979; Maida and Shapiro 1982; Shapiro and Rapaport 1986, 1987; Rapaport 1986c), is a semantic-network language with facilities for building semantic networks to represent information, for retrieving information from them, and for performing inference with them. There are at least two sorts of semantic networks in the AI literature (see Findler 1979 for a survey): The most common is what is known as an "inheritance hierarchy", of which the most well-known is probably KL-ONE (*cf.* Brachman and Schmolze 1985). In an inheritance semantic network, nodes represent concepts, and arcs represent relations between them. For instance, a typical inheritance semantic network might represent the propositions that Socrates is human and that humans are mortal as in Figure 1a. The interpreters for such systems allow properties to be "inherited", so that the fact that Socrates is mortal does not also have to be stored at the Socrates-node. What is essential, however, is that the representation of a proposition (e.g., that Socrates is human) consists only of separate representations of the individuals (Socrates and the property of being human) linked by a relation arc (the "ISA" arc). That is, propositions are not themselves objects. By contrast,

> SNePS is a *propositional* semantic network. By this is meant that all information, including propositions, "facts", etc., is represented by nodes. The benefit of representing propositions by nodes is that propositions about propositions can be represented with no limit. . . . Arcs merely form the underlying syntactic structure of SNePS. This is embodied in the restriction that one cannot add an arc between two existing nodes. That would be trantamount to telling SNePS a proposition that is not represented as a node. . . . Another restriction is the *Uniqueness Principle*: There is a one-to-one correspondence between nodes and represented concepts. This principle guarantees that nodes will be shared whenever possible and that nodes represent intensional objects. (Shapiro and Rapaport 1987.)

Thus, for example, the information represented in the inheritance network of Figure 1a could (though it need not) be represented in SNePS as in Figure 1b; the crucial difference is that the SNePS proposi-

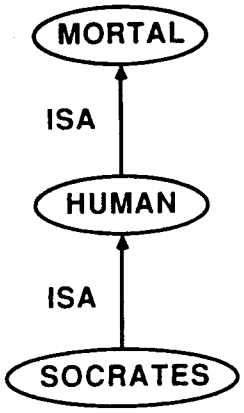

Fig. 1a. An "ISA" inheritance-hierarchy semantic network

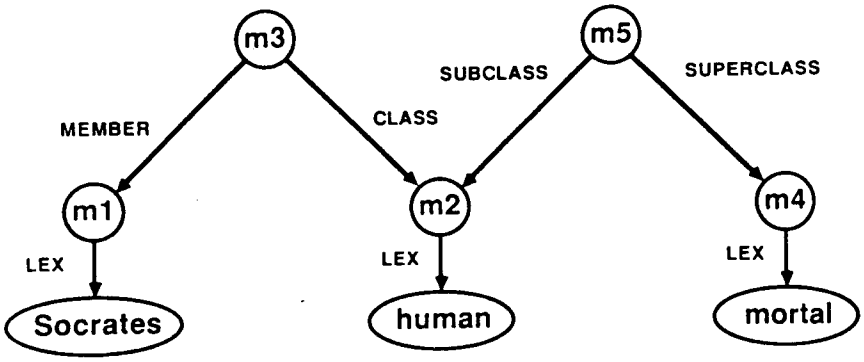

Fig. 1b. A SNePS propositional semantic network (m3 and m5 represent the propositions that Socrates is human and that humans are mortal, respectively)

tional network contains nodes (m3, m5) representing the *propositions* that Socrates is human and that humans are mortal, thus enabling representations of beliefs and rules *about* those propositions. (In fact, the network of Figure 1a could *not* be built in SNePS, by the first restriction cited; *cf.* Shapiro 1979, Section 2.3.1.) My colleagues and I in the SNePS Research Group and the Graduate Group in Cognitive Science at SUNY Buffalo are using SNePS to build a natural-language-understanding system, which we call 'CASSIE', the *C*ognitive *A*gent of the *S*NePS *S*ystem — an *I*ntelligent *E*ntity (Shapiro and Rapaport 1986, 1987; Bruder *et al.* 1986). The nodes of CASSIE's knowledge base implemented in SNePS are her beliefs and other objects of thought, in the Meinongian sense. (Needless to say, I hope, nothing

about CASSIE's *actual* state of "intelligence" should be inferred from her name!)

A brief conversation with CASSIE is presented in Appendix 1. Here, I shall sketch a small part of her natural-language-processing algorithm. Suppose that the user tells CASSIE,

Young Lucy petted a yellow dog.

CASSIE's tacit linguistic knowledge, embodied in an augmented transition network (ATN) parsing-and-generating grammar (Shapiro 1982), "hears" the words and builds the semantic network shown in Figure 2 in CASSIE's "mind" in the following way:

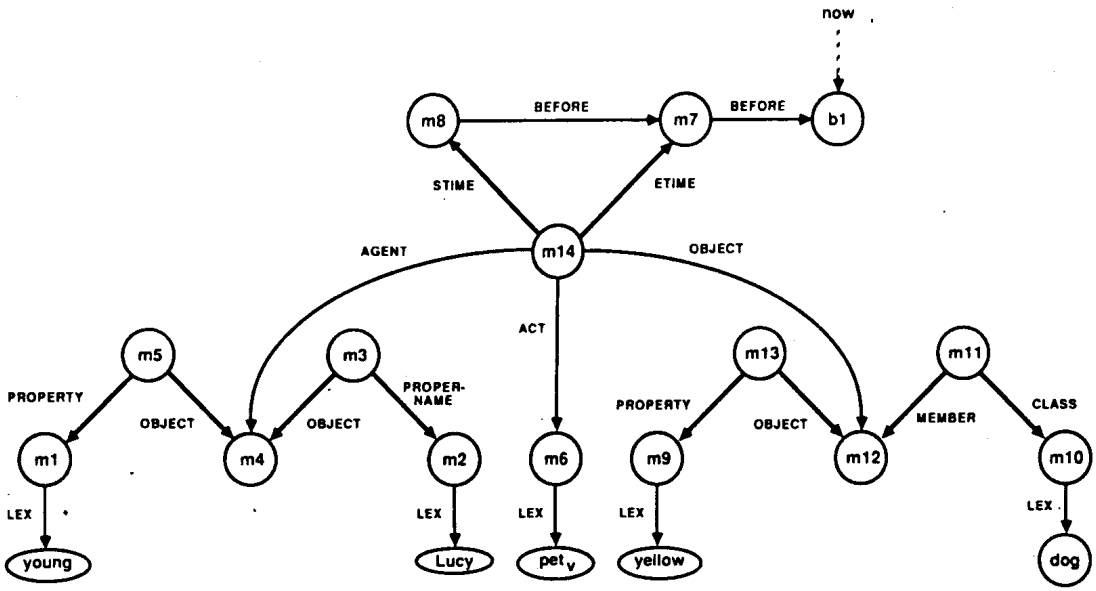

Fig. 2. CASSIE's belief that young Lucy petted a yellow dog

(1) CASSIE builds a node (b1) representing the current time (the "now"-point; *cf.* Almeida and Shapiro 1983, Almeida 1987).

(2)
- CASSIE "hears" the word 'young'.
- If she has not heard this word before, she *builds* a "sensory" node (labeled 'young') representing the *word* that she hears and a node (m1) representing the internal *concept* produced by her having heard it — this concept node is linked to the sensory node by an arc labeled 'LEX'. (See Figure 3; the formal semantic interpretation

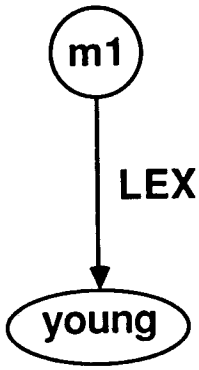

Fig. 3. SNePS network for the concept expressed in English as 'young'

of this small network is: m1 is the Meinongian object of thought corresponding to the utterance of 'young'; *cf.* Rapaport 1985a; Shapiro and Rapaport 1986, 1987.)

- If she *has* heard it before, she *finds* the already-existing concept node. (Actually, she attempts to "find" before she "builds"; henceforth, this process of "finding-or-building" will be referred to simply as "building", since it is in conformity with the Uniqueness Principle.)

(3)
- CASSIE hears the word 'Lucy'.
  She builds a sensory node (labeled 'Lucy') for the *word* 'Lucy', a concept node (m2; linked to the sensory node by a LEX arc) for the *name* 'Lucy', a concept node (m4) representing an individual, and a proposition node (m3) representing that the individual is named 'Lucy' (using an OBJECT—PROPER-NAME case frame).[12]
- She (unconsciously) determines, by means of the ATN, that Lucy is young, and she builds a proposition node (m5) representing this (using an OBJECT-PROPERTY case frame).

(4) She hears the word 'petted', and (skipping a few details, for clarity) she builds a sensory node (labeled 'pet_v') for the verb 'pet', a concept node (m6; linked to the sensory node by a LEX arc) for the act of petting, and a temporal network (m7 and m8, linked by BEFORE arcs to b1) indicating that this act occurred before "now" (= the time of utterance).

(5) She hears 'yellow' and processes it as she did 'young' (building m9).

(6) She hears 'dog' and builds:
- a sensory node for it.
- a concept node (m10) representing the class whose label is 'dog',
- a concept node (m12) representing the individual yellow dog whom young Lucy petted,
- a proposition node (m11) representing that this individual concept node is a MEMBER of the CLASS whose label is 'dog',
- a proposition node (m13) representing that that individual concept node (the dog) is yellow, and, finally,
- a proposition node (m14) representing that the dog is the OBJECT of an AGENT-ACT-OBJECT case frame whose agent is Lucy, whose act is petting, whose starting time is m8, and whose ending time is m7.

(7) She generates a sentence expressing her new understanding. I shall not go into the details of the generation algorithm, except to point out that she uses the sensory nodes to generate the words to express her new belief (*cf.* Shapiro 1982 for details).

(8) As the conversation shown in Appendix 1 continues, CASSIE's semantic network is continually updated with new nodes, as well as with new links to old nodes (*cf.* Fig. 4).

The crucial thing to see is that the semantic network (Fig. 2) that represents CASSIE's belief (the belief produced by her understanding of the user's sentence) is her interpretation of that sentence and that it has three parts: One part consists of the sensory nodes: the nodes at the heads of LEX arcs; a second part consists of the entire network except for that set of sensory nodes and the LEX arcs; and the third part consists of the LEX arcs themselves, which link the other two, major, parts of the network.

Notice that the sensory-node set by itself has (or seems to have) no structure. This is consistent with viewing these as CASSIE's internal representations, causally produced, of external entities (in this case, utterances) to which she has no other kind of access and, hence, no knowledge of their relationships. As I suggested earlier when discussing deixis, if we had a visual-input module, there might be a set of sensory nodes linked by, say, "PIX" arcs. At present, I see no need for any

direct links between visual and linguistic sensory nodes, even between those that, in some extensional sense, represent the same entity; any such links would be forged by means of links among the concept nodes at the *tails* of LEX and PIX arcs (but this is a matter for future investigation, as is the entire issue of the structure and behavior of sensory nodes).

The concept-node set, on the other hand, has a great deal of structure. It is this fragment of the entire network that represents CASSIE's internal understanding. If CASSIE were not intended to converse in natural language, there would not be any need for LEX arcs or sensory nodes. If CASSIE's sensory nodes were replaced by others, she would converse in a notational variant of English (*cf.* Quine 1969, Section II, p. 48). If her generation grammar were replaced with one for French and her sensory nodes replaced with "French" ones, she would understand English but speak in French (though here, no doubt, other modifications would be required in order for her knowledge representation system to be used in this way as an "interlingua", as machine-translation researchers call it).[13] In each of these cases, *the structure of her mind and, thus, her understanding — which would be in terms of purely syntactic symbol manipulation — would remain the same.* Only the external semantic interpretation function, so to speak, would differ. "Meaning," in the sense of internal semantics, "is determined by structures, truth by facts" (Garver 1986, p. 75).[14]

A nice metaphor for this is Carnap's example of the railroad map whose station names (but not rail-line names) have been removed; the stations can still be uniquely identified by the rail lines that meet at them. The "meaning" of a node in such a network is merely its locus in the entire network. In Appendix 2, I sketch how this might be done in a SNePS-like semantic network. (See Carnap 1928, Section 14, pp. 25-27; *cf.*: Quillian 1967, p. 101; Quillian 1968, Section 4.2.1, especially p. 238; and Quine 1951, Section 6, especially pp. 42f. Quine's "fabric which impinges on experience only along the edges" nicely captures the notion of a semantic network with sensory nodes along the edges.)

3.7. *Varieties of Meaning*

At this point, we can make the promised comments on the different kinds of meaning. Recall the three-part nature of the semantic network: the sensory nodes, the LEX arcs, and the main body of the semantic

network. The meaning of a node, in one sense of 'meaning', is its locus in the network; this is, I have been urging, the central meaning of the node. This locus provides the *internal* semantics of the node and, hence, of the words that label sensory nodes. Considered as an object of thought, a node can be taken as being constituted by a collection of properties, hence as an intensional, Meinongian object. The locus in the network of a node at the tail of a LEX arc can be taken as a collection of propositional functions corresponding to the open sentences that are satisfied by the word that labels the sensory node. In particular, at any time *t*, the collection will consist of those open sentences satisfied by the word that were heard by the system prior to *t*. (For details, see Rapaport 1981.) But this means that the internal meaning of the word will change each time the word is heard in a new sentence. So, the internal meaning is extensional. This curious duality of intension and extension is due, I think, to the fine grain of this sort of meaning: it is intensional because of its fine grain and the fact that it is an object of thought; but it is extensional in that it is determined by a set-in-extension.

But there is a meaning determined by a set-in-intension, too. This may be called the "definitional" meaning of the word. It is a subset of the internal meaning, whose characterizing feature is that it contains those propositions in the semantic network that are the meaning postulates of the word. That is, these propositions are the ones from which all other facts containing the word can be inferred (together with miscellaneous other facts; again, *cf.* Rapaport 1981). Thus, this kind of meaning is an internal, intensional meaning; it is a sort of idiosyncratic or idiolectal *Sinn*.

Both of these kinds of meaning are or consist of internal symbols to be manipulated. To fill out the picture, there may also be the (physical) objects in the world, which are the external, extensional, referential meanings of the words. But these are not symbols to be manipulated and are irrelevant for natural-language understanding.

3.8. *Discourse*

Another aspect of my interpretation of natural-language understanding is the importance of *discourse* (sequences of sentences), rather than isolated sentences, for the construction of the system's knowledge base. Discourse is important for its *cumulative* nature:

[P]utting one sentence after another can be used to express time sequence, deductive necessity, cause, exemplification or other relationships, *without any words being used to express the relation.* (Mann *et al.* 1981, Part 1, p. 6.)

This aspect of discourse illuminates the role of internal semantics in a way hinted at earlier. To provide a semantic interpretation for a language by means of an internal semantic network (or a discourse representation structure) is to provide a more or less formal *theory* about the linguistic data (much as Chomsky 1965 said, though this is a *semantic* theory). But, in discourse as in science, the data underdetermine the theory: it is internal semantic network — the mind of the understander — that provides explicit counterparts to the unexpressed relations.

Isolated sentences (so beloved by philosophers and linguists) simply would not serve for enabling a system such as CASSIE to understand natural language: they would, for all practical purposes, be random, unsystematic, and *unrelated* data. The *order* in which CASSIE processes (or "understands") sentences is important: Given a mini-discourse of even as few as two sentences,

$$s_1.\ s_2.$$

her interpretation of s_2 will be partially determined by her interpretation of s_1. Considered as part of a discourse, sentence s_2 is syntactically within the "scope" of s_1; hence, the interpretation of s_2 will be within the scope of the interpretation of s_1. (This aspect of discourse is explored in Maida and Shapiro 1982, Mann and Thompson 1983, Kamp 1984 and forthcoming, Fauconnier 1985, Asher 1986, 1987, and Wiebe and Rapaport 1986.) Thus, discourse and, hence, *conversation* are essential, the latter for important feedback in order to bring the conversers' semantic networks into alignment.

4. WOULD A COMPUTER "REALLY" UNDERSTAND?

I have considered what it would be for a computer to understand natural language, and I have argued for an interpretation of "understanding natural language" on which it makes sense to say that a computer *can* understand natural language. But there might still be some lingering doubts about whether a computer that understands natural language in this sense "really" understands it.

4.1. *The Korean-Room Argument*

Let us start with a variation of Searle's Chinese-Room Argument, which may be called the "Korean Room Argument" (though we shall do away with the room):[15]

> Imagine a Korean professor of English literature at the University of Seoul who does not understand spoken or written English but who is, nevertheless, a world authority on Shakespeare. He has established and maintains his reputation as follows: He has only read Shakespeare in excellent Korean translations. Based on his readings and, of course, his intellectual acumen, he has written, in Korean, several articles on Shakespeare's play. These articles have been translated for him into English and published in numerous, well-regarded, English-language, scholarly journals, where they have met with great success.

The Korean-Room-Argument question is this: Does the Korean scholar "understand" Shakespeare? Note that, unlike the Chinese-Room Argument, the issue is not whether he understands English; he does not. Nor does he mechanically ("unthinkingly") follow a translation algorithm; others do his translating for him. Clearly, though, he does understand Shakespeare — the literary scholarly community attests to that — and, so, he understands *something*.

Similarly, Searle in the Chinese room *can* be said to understand something, even if it isn't Chinese. More precisely (because, as I urged in Section 3.2, I don't think that Searle's Chinese-Room Argument is as precisely spelled out as it could be), an AI natural-language-understanding system can be said to understand something (or even to understand *simpliciter*), insofar as what it is doing is semantic interpretation.[16] (Of course, it does this syntactically by manipulating the symbols of its semantic interpretation.) We can actually say a bit more: it understands *natural language*, since it is a natural language that it is semantically interpreting. It is a separate question whether that which it understands is *Chinese*.[17] Now, I think there are *two* ways in which this question can be understood. In one way, it is quite obvious that if the system is understanding a natural language, then, since the natural language that it is understanding is Chinese, the system must be understanding Chinese. But in other ways, it is not so obvious. After all,

the system shares very little, if any, of Chinese culture with its interlocutors, so in what sense can it be said to "really" understand Chinese? Or in what sense can it be said to understand Chinese, as opposed to, say, code of the computer-programming language that the Chinese "squiggles" are transduced into? This Chinese-*vs.*-code issue can be resolved in favor of Chinese by the Korean-Room Argument: just as it is *Shakespeare*, not merely a Korean *translation* of Shakespeare, that the professor understands, so it is Chinese, and not the programming-language code, that Searle-in-the-room understands.

As for the cultural issue, here, I think, the answer has to be that the system understands Chinese as well as any nonnative-Chinese human speaker does (and perhaps even better than some). The only qualm one might have is that, in some vague sense, what *it* means or understands by some expression might not be what the native Chinese speaker means or understands by it. But as Quine and, later, Schank have pointed out, the same qualm can beset a conversation in our native tongue between you and me (Quine 1969, p. 46; Schank 1984, Ch. 3). As I said earlier, we systematically *mis*understand each other: we can *never* mean *exactly* what another means; but that does not mean that we cannot understand each other. We might not "really" understand each other in some deep psychological or empathic sense (if, indeed, sense can be made of that notion; *cf.* Schank 1984, pp. 44-47), but we do "really" understand each other — and the AI natural-language-understanding system can "really" understand natural language — in the only sense that matters. Two successful conversers' understandings of the expressions of their common language will (indeed, they *must*) eventually come into alignment, even if one of the conversers is a computer (*cf.* Shapiro and Rapaport 1987).

4.2. *Simon and Dreyfus vs. Winograd and SHRDLU*

The considerations thus far can help us to see what is wrong with Herbert Simon's and Hubert Dreyfus's complaints that Terry Winograd's SHRDLU program does not understand the meaning of 'own' (Winograd 1972; Simon 1977, cited in Dreyfus 1979). Simon claims that "SHRDLU's test of whether something is owned is simply whether it is tagged 'owned'. These is no intensional test of ownership . . ." (Simon 1977, p. 1064/Dreyfus 1979, p. 13). But this is simply not correct: When Winograd tells SHRDLU, "I own blocks which are not red, but I

don't own anything which supports a pyramid," he comments that these are "two new theorems . . . created for proving things about 'owning'" (Winograd 1972, p. 11, *cf.* pp. 143f; cited also in Dreyfus 1979, p. 7). SHRDLU doesn't *merely* tag blocks (although it can also do that); rather, there are procedures for determining whether something is "owned" — SHRDLU can figure out new cases of ownership.[18] So there *is* an intensional test, although it may bear little or no resemblance, except for the label 'own', to *our* intensional test of ownership. But even this claim about lack of resemblance would only hold at an early stage in a conversation; if SHRDLU were a perfect natural-language-understanding program that *could* understand English (and no one claims that it is), *eventually* its intensional test of ownership would come to resemble ours *sufficiently for us to say that it understands 'own'*.

But Dreyfus takes this one step further:

> [SHRDLU] still wouldn't understand, unless it also understood that it (SHRDLU) couldn't own anything, since it isn't a part of the community in which owning makes sense. Given our cultural practices which constitute owning, a computer cannot own something any more than a table can. (Dreyfus 1979, p. 13.)

The "community", of course, is the *human* one (which is biological; *cf.* Searle). There are several responses one can make. First of all, taken literally, Dreyfus's objection comes to nothing: it should be fairly simple to give the computer the information that, because it is not part of the right community, it cannot own anything. But that, of course, is not Dreyfus's point. His point is that it cannot *understand* 'own' because it *cannot* own. To this, there are two responses. For one thing, cultural practices can change, and, in the case at hand, they are already changing (for better or worse): computers *could* legally own things just as corporations, those other nonhuman persons, can (*cf.* Willick 1985).[19] But even if they can't, or even if there is some other relationship that computers are forever barred from participating in (even by means of a simulation), that should not prevent them from having an understanding of the concept. After all, women understood what voting was before they were enfranchised, men can understand what pregnancy is, and humans can understand what (unaided) flying is.[20] A computer could learn and understand such expressions to precisely the same extent, and that is all that is needed for it to really understand natural language.

5. DOES THE COMPUTER UNDERSTAND THAT IT UNDERSTANDS?

There are two final questions to consider. The first is this: Suppose that we have our ultimate AI natural-language-understanding program that passes the Turing Test; does it understand that it understands natural language? The second, perhaps prior, question is: *Can* it understand that it understands?

Consider a variation on the Korean-Room Argument. Suppose that the Korean professor of English literature has been systematically misled, perhaps by his translator, into thinking that the author of the plays that he is an expert on was a Korean playwright named, say, Jaegwon. The translator has systematically replaced 'Shakespeare' by 'Jaegwon', and *vice versa*, throughout all of the texts that were translated. Now, does the Korean professor understand Shakespeare? Does he understand that he understands Shakespeare? I think the answer to the latter question is pretty clearly 'No'. The answer to the former question is not so clear, but I shall venture an answer: Yes.

Before explaining this answer, let's consider another example (adapted from Goodman 1986). Suppose that a student in my Theory of Computation course is executing the steps of a Turing-machine program, as an exercise in understanding how Turing machines work. From time to time, she writes down certain numerals, representing the output of the program. Let us even suppose that they are Arabic numerals (i.e., let us suppose that she decodes the actual Turing-machine output of, say, 0s and 1s, into Arabic numerals, according to some other algorithm). Further, let us suppose that, *as a matter of fact*, each number that she writes down is the greatest common divisor of a pair of numbers that is the input to the program. Now, does she know that that is what the output is? Not necessarily; since she might not be a math major (or even a computer science major), and since the Turing-machine program need not be labeled 'Program to Compute Greatest Common Divisors', she might not know what she is doing *under that description*. Presumably, she does know what she is doing under some other description, say, "executing a Turing-machine program"; but even this is not necessary. Since, as a matter of fact, she *is* computing greatest common divisors, if I needed to know what the greatest common divisor of two numbers was, I could ask her to execute that program for me. She

would not have to understand what she is doing, under that description, in order to do it.

Similarly, the Korean professor does not have to understand that he understands Shakespeare in order to, in fact, understand Shakespeare. And, it should be clear, Searle-in-the-Chinese-room does not have to understand that he understands Chinese in order to, in fact, understand Chinese. So, a natural-language-understanding program does not have to understand that it understands natural language in order to understand natural language. That is, this use of '*understand*' is referentially transparent! If a cognitive agent A understands X, and X is equivalent to Y (in some relevant sense of equivalence), then A understands Y.

But this is only the case for "first-order" understanding: *understanding that one understands* is referentially opaque. I don't think that this is inconsistent with the transparency of first-order understanding, since this "second-order" sense of 'understand' is more akin to 'know that' or 'be aware', and the "first-order" sense of 'understand' is more akin to 'know how'.

Now, *can* the Korean professor understand that he understands Shakespeare? Of course; he simply needs to be told that it is Shakespeare (or merely someone *named* 'Shakespeare'; *cf.* Hofstadter *et al.* 1982), not someone named 'Jaegwon', that he has been studying all these years. Can my student understand that what she is computing are greatest common divisors? Of course; she simply needs to be told that. Moreover, if the program that she is executing is suitably modularized, the names of its procedures might give the game away to her. Indeed, an "automatic programming" system would have to have access to such labels in order to be able to construct a program to compute greatest common divisors (so-named or so-described); and if those labels were linked to a semantic network of mathematical concepts, it could be said to understand what that program would compute. And the program itself could understand what it was computing if it had a "self-concept" and could be made "aware" of what each of its procedures did.

This is even clearer to see in the case of a natural-language-understanding program. A natural-language-understanding program can be made to understand what it is doing — can be made to understand that it understands natural language — by, first, telling it (in natural language, of course) that that is what it is doing. Merely telling it, however, is not sufficient by itself; that would merely add some network structure to its knowledge base. To gain the requisite "aware-

ness", the system would have to have LEX-like arcs linked, if only indirectly, to its actual natural-language-processing module — the ATN parser-generator, for instance. But surely that can be done; it is, in any event, an empirical issue as to precisely how it would be done. The point is that it would be able to associate expressions like 'understanding natural language' with certain of its activities. It would then understand what those expressions meant in terms of what those activities were. It would not matter in the least if it understood those activities in terms of bit-patterns or in terms of concepts such as "parsing" and "generating"; what would count is this: that it understood the expressions in terms of its actions; that its actions were, in fact, actions for understanding natural language; and, perhaps, that 'understanding natural language' was the label that its interlocutors used for that activity.

6. CONCLUSION

By way of conclusion, consider (a) the language L that the system understands, (b) the external world, W, about which L expresses information, and (c) the language (or model of W), L_w, that provides the interpretation of L. As William A. Woods (among many others) has made quite clear, such a "meaning representation language" as L_w is involved in two quite separate sorts of semantic analyses (Woods 1978; *cf.* Woods 1975 and, especially, Kamp 1981).

There must be, first, a semantic interpretation function, P (for 'parser'), from utterances of L (the input to the system) to the system's internal knowledge base, L_w. L_w is the system's model of W, as filtered through L. There will also need to be a function, G (for 'generator'), from L_w to L, so that the system can express itself. P and G need not, and probably should not, be inverses (they are not in SNePS/CASSIE); "they" might also be a single function (as in SNePS/CASSIE; *cf.* Shapiro 1982). Together, P, G, and L_w constitute the central part of the *system's* understanding of L.

But, second, there must also be a semantic interpretation of L_w in terms of W (or in terms of *our* idiosyncratic L_ws) — i.e., a semantic interpretation of the domain of semantic interpretation. Since L_w is itself a formal language, specified by a formal syntax, it needs a semantics (*cf.* Woods 1975, McDermott 1981). But this semantic interpretation is merely *our* understanding of L_w. It is independent of

and external to the relevant semantic issue of how the *system* understands L. (This semantic interpretation of the knowledge base is provided for SNePS/CASSIE by interpreting L_w as a Meinongian theory of the objects of thought; *cf.*: Rapaport 1985a; Shapiro and Rapaport 1986, 1987.)

There may be another relationship between L_w and W, although this may also be provided by the semantic interpretation of L_w. This relationship is the causal one from W to L_w, and there is no reason to hold that it is limited to humans (or other biological entities). It produces the sensory nodes, but — other than that — it is also independent of and external to the system's understanding of L.

Once the sensory and concept nodes (or their analogues in some other knowledge-representation system) are produced, the actual causal links cease to be relevant to the system's *understanding* (except — and I am willing to admit that this is an important exception — for purposes of the system's communication with others), thus highlighting the representationalism of the system.

Searle holds, however, that the links — the access to W — are necessary for understanding, that humans have (or that only biological entities can have) such access, that computers lack it, and, hence, that computers cannot understand. By contrast, I hold that *if* such access *were* needed, then computers could have it, too, so that Searle's pessimism with respect to computer understanding is unsupported. I also hold that such access is *not* needed, that, therefore, humans don't need it either (here is where methodological solipsism appears), so that, again, there's no support for Searle's conclusion. I agree with Searle that semantics is necessary for understanding natural language, but that the *kind* of semantics that's needed is the semantics provided by an internal semantic interpretation, which is, in fact, syntactic in nature and, hence, computable. Syntax suffices.

APPENDIX 1: A "CONVERSATION" WITH CASSIE

Following is the transcript of a "conversation" with CASSIE. A commented version of part of it appears in Shapiro and Rapaport 1986, 1987. User input is on lines with the :-prompt; CASSIE's output is on the lines that follow. A fragment of the full network showing CASSIE's state of mind at the end of the conversation is shown in Figure 4.

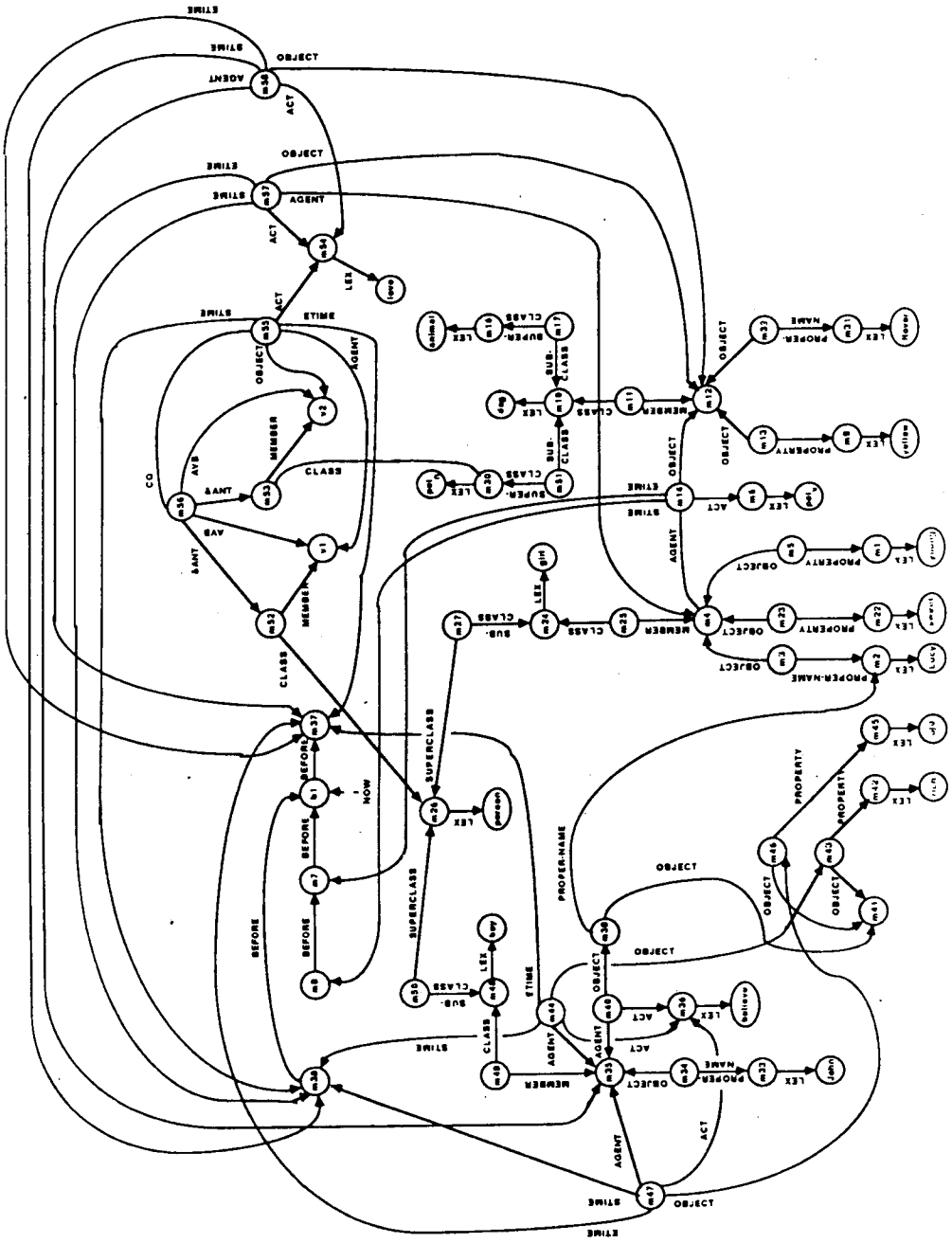

Fig. 4. Small fragment of CASSIE's beliefs at the end of the conversation.

: Young Lucy petted a yellow dog
I understand that young Lucy petted a yellow dog

: What is yellow
a dog is yellow

: Dogs are animals
I understand that dogs are animals

: Who petted an animal
young Lucy petted a yellow dog

: Lucy is clever
I understand that young Lucy is clever

: Lucy is a girl
I understand that clever young Lucy is a girl

: Girls are people
I understand that girls are people

: What was petted by a person
a yellow dog was petted by clever young Lucy

: The dog is named Rover
I understand that Rover is a yellow dog

: What was petted by a person
yellow Rover was petted by clever young Lucy

: For every d if d is a dog then there is a b such that b is a boy and b owns d
I understand that for every d, if d is a dog
then there exists a b such that b is a boy
and
b owns d

: The boy is named John
I understand that John is a boy

: Who owns Rover
John owns yellow Rover

: John believes that Lucy is rich
I understand that John believes that Lucy is rich

: Who is rich
I don't know.

: Who is clever
young Lucy is clever

: John believes that Lucy is old
I understand that John believes that rich Lucy is old

: Boys are people
I understand that boys are people

: Dogs are pets
I understand that dogs are pets

: For every p and d if p is a person and d is a pet then p loves d
I understand that for every d and p, if p is a person
and
d is a pet
then p loves d

: Who loves a pet
clever young Lucy loves yellow Rover
and
John loves yellow Rover

APPENDIX 2: DESCRIBING A NODE VIA ITS NETWORK LOCUS

How can a node be identified if there are no LEX arcs or sensory nodes? That is, how can they be identified if they have no names? The answer is, by descriptions. It is important to see that the identifiers of the nodes ("m1", etc.) are *not* names (or labels); they convey no information to the system. (They are like the nodes of a tree each of which contains no data but only pointers to its left and right children. The sensory nodes are like leaf nodes that do contain data; their labels do convey information.) The nodes can be described solely in terms of their locus in the network, i.e., in terms of the structure of the arcs (which *are* labeled) that meet at them. If a node has a unique "arc structure", then it can be uniquely described by a *definite* description; if two or more nodes share an arc-structure, they can only be given *indefinite* descriptions and, hence, cannot be uniquely identified. That

is, they are indistinguishable to the system, unless each has a LEX arc emanating from it. (*Cf.*, again, Carnap 1928, Section 14.) Thus, for example, in the network in Figure 5, m1 is *the* node with precisely two ARG arcs emanating from it, and b1 is *a* node with precisely one ARG arc entering it (and similarly for b2). In keeping with the notion that the internal meaning of a node is its locus in the *entire* network, the *full* descriptions of m1 and b1 (or b2) are:

(m1) *the* node with one ARG arc to *a* base node and with another ARG arc to *a* base node.

(b1) *a* base node with an ARG arc from *the* node with one ARG arc to it and with another ARG arc to a base node.

(A base node is a node with no arcs leaving it; no SNePS node can have an arc pointing to itself.) The pronominal 'it' has widest scope; i.e., its anaphoric antecedent is always the node being described. Note that each node's description is a monad-like description of the *entire* network from its own "point of view".

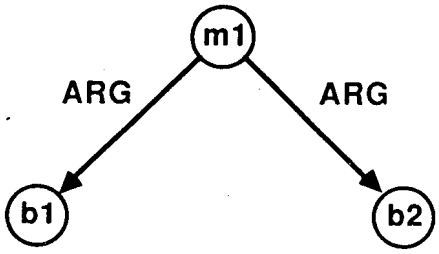

Fig. 5. A small SNePS network.

NOTES

[1] This material is based upon work supported by the National Science Foundation under Grant Nos. IST-8504713, IRI-8610517, and by SUNY Buffalo Research Development Fund Award No. 150-8537-G. I am grateful to Randall R. Dipert, Michael Leyton, Ernesto Morgado, Jane Pease, Sandra Peters, Stuart C. Shapiro, Marie Meteer Vaughan, Janyce M. Wiebe, Albert Hanyong Yuhan, and other colleagues in the SNePS Research Group and the SUNY Buffalo Graduate Group in Cognitive Science for discussions on these topics and comments on earlier versions of this essay.

[2] *Cf.* my earlier critiques of Searle, in which I distinguish between an abstraction, an implementing medium, and the implemented abstraction (Rapaport 1985b, 1986b, and forthcoming).

[3] The "weak/strong" terminology is from Searle 1980.

[4] Or: that it is a woman. More precisely. Turing describes the Imitation Game, in which "a man (A), a woman (B), and an interrogator (C)" have as their object "for the interrogator to determine which of the other two is the man and which is the woman". Turing then modifies this:

> We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?" (Turing 1950, p. 5.)

[5] Randall R. Dipert has suggested to me that Searle's Chinese-Room Argument does show that what *executes* the program (viz., the central processing unit) does not understand, leaving open the question whether the *process* might understand.

[6] Not to be confused with scripts in the sense of Schank's AI data structures.

[7] Similarly, part of my argument in this essay may be roughly paraphrased as follows: I want to understand what it means to understand natural language; I believe that it is capable of being understood (that it is not a mystery) and that, for a system to understand natural language, certain formal techniques are necessary and sufficient; these techniques are computational; hence, understanding natural language is a recursive function.

[8] I take (7) to be the conclusion, since (1)—(6) are in response to the question, "*Can* computers add?" (p. 25; italics added). If I am taking Dretske too literally here, then simply end the argument at step (6).

[9] Note that, for independent reasons, the computer *will* need an internal representation or model of itself. Maybe this won't be a *complete* self-model, on pain of infinite regress, but then neither is ours. If needed, it, and we, can use an *external* model that *is* complete, via self-reflection; *cf.* Case 1986, esp. p. 91. For more on self-models, *cf.*: Minsky 1965; Rapaport 1984, 1986c; Smith 1986.

[10] I venture to say that the mutual learning and adjusting process has already begun: studying such computers and primitive AI systems as we have now has led many philosophers and AI researchers to this kind of opinion.

[11] It should be obvious by now that by 'understand' I do not mean some sort of "deep" psychological understanding, merely that sort of understanding required for understanding natural language. *Cf.* Schank 1984, Ch. 3.

[12] *Cf.* Shapiro and Rapaport 1986 and 1987 for the formal syntax and semantics of this and the other case frames. The node identifiers ("m1", etc.) are generated by the underlying program in an implementation-dependent order; the order and the identifiers are inessential to the semantic network.

[13] The importance of the knowledge base, whether it is a semantic network, a discourse representation structure, or some other data type, for understanding natural language has some interesting implications for machine translation. There are several paradigms for machine translation; two are relevant for us: the "transfer" approach and the "interlingua" approach (*cf.* Slocum 1985). Transfer approaches typically do not use a knowledge base, but manipulate syntactic structures of the source language until they turn into syntactic structures of the target language. Such a system, I would argue, cannot be said to understand the natural languages it deals with. Interlingua approaches, on the other hand, do have a sort of knowledge base. They are "mere" symbol manipulation systems, but the symbols that get manipulated include those of the

system's internal knowledge-representation system: hence, interlingua machine-translation systems have at least the *potential* for understanding. (Searle's Chinese-language program appears to be more like a transfer system (for, say, translating Chinese into Chinese) than an interlingua system, despite the use of Schank-like scripts.)

Note, too, that this suggests that the "machine-translation problem" is coextensive with the "natural-language-understanding problem" and, thus (*cf.* Section 1, above), with the general "AI problem": solve one and you will have solved them all. (This underlies Martin Kay's pessimism about the success of machine translation; *cf.* Kay 1986).

[14] Garver's "challenge of metaphor", it must be noted, is also a challenge for the theory presented here, which I hope to investigate in the future.

[15] The Korean-Room Argument was suggested to me by Albert Hanyong Yuhan.

[16] And to the extent that it does *not* do semantic interpretation, it does not understand. My former teacher, Spencer Brown, recently made the following observation:

> As for Searle, I myself have been a *corpus vile* for his "experiment": once I conveyed a message from one mathematician to another, with complete understanding on the part of the second and with total, nay, virginal, ignorance on my part of the meaning of the message. Similarly I have conveyed a message from my doctor to my dentist without knowing what I was telling. Q.E.D.: I can't think. This is something I have always suspected. (Personal communication, 1986; *cf.* Rapaport 1981, p. 7.)

But the conclusions (both of them!) are too hasty: all that follows is that he did not understand certain isolated statements of mathematics and medicine. And this was, no doubt, because he lacked the tools for interpreting them and an appropriate knowledge base within which to fit them.

[17] I owe this way of looking at my argument to Michael Leyton.

[18] I am indebted to Stuart C. Shapiro for pointing this out to me.

[19] I am indebted to Shapiro for this reference.

[20] The examples are due to Shapiro. My original example was that, as a U.S. citizen, I am probably forever enjoined from some custom unique to and open only to French citizens; yet surely I can learn and understand the meaning of the French expression for such a custom. But finding an example of such a custom is not as easy as it seems. Voting in a French election, e.g., isn't quite right, since I *can* vote in U.S. elections, and similarly for other legal rights or proscriptions. Religious practices "unique" to one religion usually have counterparts in others. Another kind of case has to do with performatives: I cannot marry two people merely by reciting the appropriate ritual, since I do not have the right to do so. The case of owning that Simon and Dreyfus focus on is somewhat special, since it is both in the legal realm as well as the cultural one: there are (allegedly or at least conceivably) cultures in which the institutions of ownership and possession are unknown. I maintain, however, that the case of humans and computers are parallel: we share the same abilities and inabilities to understand or act.

REFERENCES

Allen, Woody: 1980, 'The Kugelmass Episode' in W. Allen, *Side Effects*, Random House, New York, 41—55.

Almeida, Michael J.: 1987, 'Reasoning about the Temporal Structure of Narratives', Technical Report 86—10, Department of Computer Science, Buffalo, SUNY Buffalo.

Almeida, Michael J., and Shapiro, Stuart C.: 1983, 'Reasoning about the Temporal Structure of Narrative Texts', *Proc. Fifth Annual Meeting of the Cognitive Science Society*, Rochester, N.Y., unpaginated.

Appelt, Douglas E.: 1982, 'Planning Natural-Language Utterances', *Proc. of the National Conference on Artificial Intelligence (AAAI-82; Pittsburgh)*, Morgan Kaufmann, Los Altos, CA, pp. 59—62.

Appelt, Douglas E.: 1985, 'Some Pragmatic Issues in the Planning of Definite and Indefinite Noun Phrases', *Proc. 23rd Annual Meeting of the Assoc. for Computational Linguistics (University of Chicago)*, Assoc. for Computational Linguistics, Morristown, N.J.; pp. 198—203.

Asher, Nicholas: 1986, 'Belief in Discourse Representation Theory', *Journal of Philosophical Logic* **15**, 127—89.

Asher, Nicholas: 1987, 'A Typology for Attitude Verbs and Their Anaphoric Properties', *Linguistics and Philosophy* **10**, 125—197.

Brachman, Ronald J., and Levesque, Hector J.: 1985, *Readings in Knowledge Representation*, Morgan Kaufmann, Los Altos, CA.

Brachman, Ronald J., and Schmolze, James G.: 1985, 'An Overview of the KL-ONE Knowledge Representation System', *Cognitive Science* **9**, 171—216.

Bruder, Gail A., Duchan, Judith F., Rapaport, William J., Segal, Erwin M., Shapiro, Stuart C., and Zubin, David A.: 1986, 'Deictic Centers in Narrative: An Interdisciplinary Cognitive-Science Project', Technical Report No. 86—20, Dept. of Computer Science, SUNY Buffalo, Buffalo.

Carnap, Rudolf: 1928, *The Logical Structure of the World*, R. A. George (trans.), Univ. of California Press, Berkeley, 1967.

Case, John: 1986, 'Learning Machines', in W. Demopoulos and A. Marras (eds.), *Language Learning and Concept Acquisition*, Ablex, Norwood, N.J., 83—102.

Cohen, Philip R., & Perrault, C. Raymond: 1979, 'Elements of a Plan-Based Theory of Speech Acts', *Cognitive Science* **3**, 177—212; reprinted in B. L., Webber & N.J. Nilsson (eds.), *Readings in Artificial Intelligence* Tioga Publishing Co., Palo Alto, pp. 478—95.

Davies, Robertson: 1972, *The Manticore*, In R. Davies, *The Deptford Trilogy*, Penguin, Middlesex, Eng. 1983.

de Saussure, Ferdinand: 1915, *Cours de linguistique générale*, Payot, Paris, 1972.

Dennett, Daniel C.: 1983, 'Intentional Systems in Cognitive Ethology: The 'Panglossian Paradigm' Defended', *Brain and Behavioral Sciences* **6**, 343—390.

Dretske, Fred: 1985, 'Machines and the Mental', *Proc. and Addresses of the American Philosophical Assoc.* **59**, 23—33.

Dreyfus, Hubert L.: 1979, *What Computers Can't Do: The Limits of Artificial Intelligence*, Harper & Row, New York, rev. ed.; Introduction to the Revised Edition, pp. 1—87; reprinted with revisions in Haugeland 1981, pp. 161—204, and in Brachman and Levesque 1985, pp. 71—93. Page references are to Dreyfus's book.

Fauconnier, Gilles: 1985, *Mental Spaces: Aspects of Meaning Construction in Natural Language*, MIT Press, Cambridge, MA.

Findler, Nicholas V.: 1979, *Associative Networks: The Representation and Use of Knowledge by Computers*, Academic Press, New York.

Garver, Newton: 1986, 'Structuralism and the Challenge of Metaphor', *Monist* **69**, 68—86.

Gleick, James: 1985, 'They're Getting Better about Predicting the Weather (Even Though You Don't Believe It)', *The New York Times Magazine*, 27 January.

Goodman, Nicolas D.: 1986, 'Intensions, Church's Thesis, and the Formalization of Mathematics', *Notre Dame Journal of Formal Logic* (forthcoming).

Green, Bert F., Wolf, Alice K., Chomsky, Carol, and Laughery, Kenneth: 1961, 'Baseball: An Automatic Question Answerer', reprinted in E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*, McGraw-Hill, New York, 1963, pp. 207—16.

Haugeland, John (ed.): 1981, *Mind Design: Philosophy, Psychology, Artificial Intelligence* MIT Press, Cambridge, MA.

Hofstadter, Douglas R.: 1980, 'Reductionism and Religion', *Behavioral and Brain Sciences* **3**, 433—34.

Hofstadter, Douglas R, Clossman, Gray A., and Meredith, Marsha J.: 1982, 'Shakespeare's Plays Weren't Written by Him, but by Someone Else of the Same Name: An Essay on Intensionality and Frame-Based Knowledge Representation Systems', Indiana University Linguistics Club, Bloomington, IN.

Jackendoff, Ray: 1985, 'Information Is in the Mind of the Beholder', *Linguistics and Philosophy* **8**, 23—34.

Jennings, Richard C.: 1985, 'Translation, Interpretation and Understanding', paper read at the American Philosophical Assoc. Eastern Division (Washington, DC); abstract, *Proc. and Addresses of the American Philosophical Assoc.* **59**, 345—46.

Kamp, Hans: 1984, 'A Theory of Truth and Semantic Representation', in J. Groenendijk, T. M. V. Janssen, and M. Stokhof (eds.), *Truth, Interpretation and Information* Foris, Dordrecht, pp. 1—41.

Kamp, Hans: (forthcoming), *Situations in Discourse*, manuscript, Center for Cognitive Science, University of Texas, Austin, TX.

Kay, Martin: 1986, 'Forum on Machine Translation: Machine Translation will not Work', *Proc. 24th Annual Meeting of the Association for Computational Linguistics (Columbia University)*, Assoc. for Computational Linguistics, Morristown, NJ, p. 268.

Lehnert, W. G., M. G. Dyer, P. N. Johnson, C. J. Yang and S. Harley: 1983, 'BORIS—An Experiment in In-Depth Understanding of Narratives', *Artificial Intelligence* **20**, 15—62.

Levesque, Hector J.: 1984, 'A Logic of Implicit and Explicit Belief', *Proc. of the National Conference on Artificial Intelligence (AAAI-84; Austin, TX)*, Morgan Kaufmann, Los Altos, CA, pp. 198—202.

Maida, Anthony S., and Shapiro, Stuart C.: 1982, 'Intensional Concepts in Propositional Semantic Networks', *Cognitive Science* **6**, 291—330; reprinted in Brachman and Levesque 1985, pp. 169—89.

Mann, William C., Bates, Madeline, Grosz, Barbara J., McDonald, David D., McKeown, Kathleen R., and Swartout, William R.: 1981, 'Text Generation: The State of the Art and the Literature', Technical Report No. ISI/RR-81-101, Information Sciences Institute, Univ. of Southern California, Marina del Rey, CA.

Mann, William C., and Thompson, Sandra S.: 1983, 'Relational Propositions in Discourse', Technical Report No. ISI/RR-83-115, Information Sciences Institute, Univ. of Southern California, Marina del Rey, CA.

McDermott, Drew: 1981, 'Artificial Intelligence Meets Natural Stupidity', in Haugeland 1981, pp. 143—60.

Minsky, Marvin L.: 1965, 'Matter, Mind, and Models', in Minsky 1968, pp. 425—32.

Minsky, Marvin (ed.): 1968, *Semantic Information Processing*, MIT Press, Cambridge, MA.

Minsky, Marvin: 1975, 'A Framework for Representing Knowledge', in Haugeland 1981, pp. 95—128; reprinted in Brachman and Levesque 1985, pp. 245—62.

Neal, Jeannette G.: 1981, 'A Knowledge Engineering Approach to Natural Language Understanding', Technical Report No. 179, Dept. of Computer Science, SUNY Buffalo, Buffalo.

Neal, Jeannette G.: 1985, 'A Knowledge Based Approach to Natural Language Understanding', Technical Report No. 85—06, Dept. of Computer Science, SUNY Buffalo, Buffalo.

Neal, Jeannette G., and Shapiro, Stuart C.: 1984, 'Knowledge Based Parsing', Technical Report No. 213, Dept. of Computer Science, SUNY Buffalo, Buffalo.

Neal, Jeannette G., and Shapiro, Stuart C.: 1985, 'Parsing as a Form of Inference in a Multiprocessing Environment', *Proc. Conf. on Intelligent Systems and Machines* Oakland University, Rochester, MI, pp. 19—24.

Neal, Jeannette G., and Shapiro, Stuart C.: 1987, 'Knowledge Representation for Reasoning about Language', in J. C. Bouderaux *et al.* (eds.), *The Role of Language in Problem Solving* **2**, Elsevier/North-Holland, pp. 27—46.

Putnam, Hilary: 1975, 'The Meaning of "Meaning"', reprinted in H. Putnam, *Mind, Language and Reality*, Cambridge University Press, Cambridge Eng, pp. 215—71.

Quillian, M. Ross: 1967, 'Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities', *Behavioral Science* **12**, 410—30; reprinted in Brachman and Levesque 1985, pp. 97—118. Page references are to the reprint.

Quillian, M. Ross: 1968, 'Semantic Memory', in Minsky 1968, pp. 227—70.

Quine, Willard Van Orman: 1951, 'Two Dogmas of Empiricism', reprinted in W. V. O. Quine, *From a Logical Point of View*, Harvard University Press, Cambridge MA, 2nd ed., revised, 1980, pp. 20—46.

Quine, Willard Van Orman: 1969, 'Ontological Relativity', in W. V. O. Quine, *Ontological Relativity and Other Essays* Columbia University Press, New York, pp. 26—68.

Rapaport, William J.: 1976, *Intentionality and the Structure of Existence*. Ph.D. dissertation, Dept. of Philosophy, Indiana University, Bloomington, IN.

Rapaport, William J.: 1981, 'How to Make the World Fit Our Language: An Essay in Meinongian Semantics', *Grazer Philosophische Studien* **14**, 1—21.

Rapaport, William J.: 1984, 'Belief Representation and Quasi-Indicators', Technical Report No. 215, Department of Computer Science, SUNY Buffalo, Buffalo.

Rapaport, William J.: 1985a, 'Meinongian Semantics for Propositional Semantic Networks', *Proc. 23rd Annual Meeting Assoc. for Computational Linguistics (University of Chicago)*, Assoc. for Computational Linguistics, Morristown, N.J. pp. 43—48.

Rapaport, William J.: 1985b, 'Machine Understanding and Data Abstraction in Searle's Chinese Room', *Proc. 7th Annual Meeting Cognitive Science Soc. (University of California at Irvine)*, Lawrence Erlbaum, Hillsdale, N.J. pp. 341—45.

Rapaport, William J.: 1985/1986, 'Non-Existent Objects and Epistemological Ontology', *Grazer Philosophische Studien* **25/26**, 61—95.

Rapaport, William J.: 1986a, 'Searle's Experiments with Thought', *Philosophy of Science* 53: 271—279; preprinted as Technical Report 216, Dept. of Computer Science, SUNY Buffalo, Buffalo, 1984.

Rapaport, William J.: 1986b, 'Philosophy, Artificial Intelligence, and the Chinese-Room Argument', *Abacus* **3** (Summer 1986), 6—17.

Rapaport, William J.: 1986c, 'Logical Foundations for Belief Representation', *Cognitive Science* **10**, 371—422.

Rapaport, William J.: (forthcoming), 'To Think or Not to Think', *Noûs*.

Rapaport, William J.: 1987, 'Belief Systems', in S. C. Shapiro (ed.), *Encyclopedia of Artificial Intelligence*, John Wiley, New York, pp. 63—73.

Russell, Bertand: 1918, 'The Philosophy of Logical Atomism', in B. Russell, *Logic and Knowledge: Essays 1901—1950*, R. C. Marsh (ed.), Capricorn, New York, 1956, pp. 177—281.

Sayre, Kenneth, M.: 1986, 'Intentionality and Information Processing: An Alternative Model for Cognitive Science', *Behavioral and Brain Sciences* **9**, 121—166.

Schank, Roger C.: 1982, *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*, Cambridge University Press, Cambridge, Eng.

Schank, Roger C. (with Childers, Peter G.): 1984, *The Cognitive Computer: On Language, Learning, and Artificial Intelligence*, Addison-Wesley, Reading, MA.

Searle, John R.: 1980, 'Minds, Brains, and Programs', *Behavioral and Brain Sciences* **3**, 417—57.

Searle, John R.: 1982, 'The Myth of the Computer', *New York Review of Books*, 29 April, 3—6; *cf.* correspondence, same journal, 24 June 1982, 56—57.

Searle, John R.: 1984, *Minds, Brains and Science*, Harvard University Press, Cambridge, MA.

Shapiro, Stuart C.: 1977, 'Representing Numbers in Semantic Networks: Prolegomena', *Proc. 5th International Joint Conference on Artificial Intelligence (IJCAI-77; MIT)*, Morgan Kaufmann, Los Altos, CA, p. 284.

Shapiro, Stuart C.: 1979, 'The SNePS Semantic Network Processing System', in Findler 1979, pp. 179—203.

Shapiro, Stuart C.: 1982, 'Generalized Augmented Transition Network Grammars For Generation From Semantic Networks', *American Journal of Computational Linguistics* **8**, 12—25.

Shapiro, Stuart C., and Rapaport, William J.: 1986, 'SNePS Considered as a Fully Intensional Propositional Semantic Network', *Proc. National Conference on Artificial Intelligence (AAAI-86; Philadelphia)*, Vol. 1, Morgan Kaufmann, Los Altos, CA, pp. 278—83.

Shapiro, Stuart C., and Rapaport, William J.: 1987, 'SNePS Considered as a Fully Intensional Propositional Semantic Network', in G. McCalla and N. Cercone (eds.), *The Knowledge Frontier*, Springer-Verlag, Berlin.

Simon, Herbert A.: 1977, 'Artificial Intelligence Systems that Can Understand', *Proc. of the 5th International Joint Conference on Artificial Intelligence (IJCAI-77; MIT)*, Morgan Kaufmann, Los Altos, CA, pp. 1059—73.

Slocum, Jonathan: 1985, 'A Survey of Machine Translation: its History, Current Status, and Future Prospects', *Computational Linguistics* **11**, 1—17.

Smith, Brian C.: 1982, 'Prologue to "Reflection and Semantics in a Procedural Language', in Brachman and Levesque 1985, 31—39.

Smith, Brian C.: 1986, 'Varieties of Self-Reference', In J. Y. Halpern (ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. of the 1986 Conference*, Morgan Kaufmann, Los Altos, CA, pp. 19—43.

Tanenbaum, Andrew S.: 1976, *Structured Computer Organization*, Prentice-Hall, Englewood Cliffs, N.J.

Turing, Alan M.: 1950, 'Computing Machinery and Intelligence', *Mind* **59**; reprinted in A. R. Anderson (ed.), *Minds and Machines*, Prentice-Hall, Englewood Cliffs, N.J., 1964, pp. 4—30.

Weizenbaum, Joseph: 1966, 'ELIZA—A Computer Program for the Study of Natural Language Communcation between Man and Machine', *Communications of the Association for Computing Machinery* **9**, 36—45. Reprinted in *CACM* **26**(1983), 23—28.

Weizenbaum, Joseph: 1974, 'Automating Psychotherapy', *ACM Forum* **17**, 543; reprinted with replies, *CACM* **26**(1983), 28.

Weizenbaum, Joseph: 1976, *Computer Power and Human Reason: From Judgment to Calculation*, W. H. Freeman, San Francisco.

Wiebe, Janyce M., and Rapaport, William J.: 1986, 'Representing *De Re* and *De Dicto* Belief Reports in Discourse and Narrative', *Proc. of the IEEE*, Special Issue on Knowledge Representation, **74**, 1405—1413.

Willick, Marshal S.: 1985, 'Constitutional Law and Artificial Intelligence: The Potential Legal Recognition of Computers as "Persons", *Proc. of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85; Los Angeles)*, Morgan Kaufmann, Los Altos, CA, pp. 1271—73.

Winograd, Terry: 1972, *Understanding Natural Language*, Academic Press, Orlando, FL.

Woods, William A.: 1975, 'What's in a Link: Foundations for Semantic Networks', in D. G. Bobrow and A. M. Collins (eds.), *Representation and Understanding*, Academic Press, New York, pp. 35—82. Reprinted in Brachman and Levesque 1985, pp. 217—41.

Woods, William A.: 1978, 'Semantics and Quantification in Natural Language Question Answering', in M. C. Yovits (ed.), *Advances in Computers*, Vol. 17, Academic Press, New York, pp. 1—87.

Department of Computer Science, and
Graduate Group in Cognitive Science
State University of New York at Buffalo
Buffalo, NY 14260, U.S.A.

SYNTACTIC SEMANTICS: FOUNDATIONS OF COMPUTATIONAL NATURAL-LANGUAGE UNDERSTANDING

In James H. Fetzer (ed.), *Aspects of Artificial Intelligence*
(Dordrecht, Holland: Kluwer Academic Publishers, 1988): 81–131.

Reprinted in:
Eric Dietrich (ed.), *Thinking Computers and Virtual Persons: Essays on the Intentionality of Machines*
(San Diego: Academic Press, 1994): 225–273.

William J. Rapaport

Dept. of Computer Science & Engineering and Center for Cognitive Science
State University of New York at Buffalo, Buffalo, NY 14260
rapaport@cse.buffalo.edu, http://www.cse.buffalo.edu/~rapaport

ERRATA

LOCATION	ERROR	CORRECTION
p. 83, line –8	repidly	rapidly
p. 89, line 5	to	the
p. 89, line 7	the	to
p. 100, line –5	would	world
p. 113, line 10	internal	the internal
p. 114, line 12	play	plays
p. 115, line –3	These	There
p. 119, §6, line 7	Kamp 1981	Kamp 1984
p. 127, after Case 1986	*missing reference*	Chomsky, Noam (1965), *Aspects of the Theory of Syntax* (Cambridge, MA: MIT Press).
p. 129, Neal & Shapiro 1987	Bouderaux	Boudreaux

UPDATES

1. p. 128: Goodman, forthcoming =

   Goodman, Nicolas (1986), "Intensions, Church's Thesis, and the Formalization of Mathematics", *Notre Dame Journal of Formal Logic* 28: 473–489.

2. p. 128: Kamp, forthcoming =

   Kamp, Hans (1983), "Situations in Discourse" (Austin: University of Texas Center for Cognitive Science).

3. p. 130: Rapaport, forthcoming =

   Rapaport, William J. (1988), "To Think or Not to Think", *Noûs* 22: 585–609.

4. p. 130: Shapiro & Rapaport 1987 =

   Shapiro, Stuart C., & Rapaport, William J. (1987), "SNePS Considered as a Fully Intensional Propositional Semantic Network", in Nick Cercone & Gordon McCalla (eds.), *The Knowledge Frontier: Essays in the Representation of Knowledge* (New York: Springer-Verlag): 262–315.