WHAT'S IN A LINK: FOUNDATIONS FOR SEMANTIC NETWORKS

W. A. Woods

Bolt Beranek and Newman, Incorporated

Prepared for:

Office of Naval Research

November 1975

BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

BBN Report No. 3072
A.I. Report No. 38

WHAT'S IN A LINK:

Foundations for Semantic Networks

W. A. Woods

November 1975

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 2904

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0533.

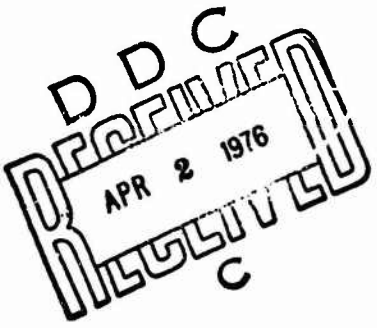
The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

What's in a Link:

Foundations for Semantic Networks

W.A. Woods

Bolt Beranek and Newman Inc.



To Appear in:

Bobrow & Collins (eds.) Representation and Understanding: Studies in Cognitive Science, New York: Academic Press (1975).

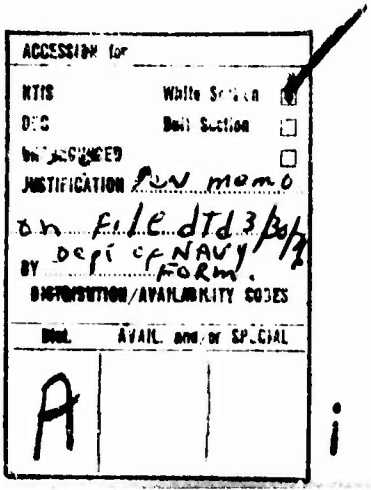


TABLE OF CONTENTS

I. Introduction

This paper is concerned with the theoretical underpinnings for semantic network representations of the sort dealt with by Quillian [1966,1969], Rumelhart, Lindsay, & Norman [1972], Carbonell & Collins [1973], Schank [1975], Simmons [1973], etc. (I include Schank's conceptual dependency representations in this class although he himself may deny the kinship). I am concerned specifically with understanding the semantics of the semantic network structures themselves, i.e., with what the notations and structures used in a semantic network can mean, and with interpretations of what these links mean that will be logically adequate to the job of representing knowledge. I want to focus on several issues: the meaning of "semantics", the need for explicit understanding of the intended meanings for various types of arcs and links, the need for careful thought in choosing conventions for representing facts as assemblages of arcs and nodes, and several specific difficult problems in knowledge representation - especially problems of relative clauses and quantification.

I think we must begin with the realization that there is currently no "theory" of semantic networks. The notion of semantic networks is for the most part an attractive notion which has yet to be proven. Even the question of what networks have to do with semantics is one which takes

some answering. I am convinced that there is real value to the work that is being done in semantic network representations and that there is much to be learned from it. However, I feel the major discoveries are yet to be made and what is currently being done is not really understood. In this paper, I would like to make a start at such an understanding.

I will attempt to show that when the semantics of the notations are made clear, many of the techniques used in existing semantic networks are inadequate for representing knowledge in general. By means of examples, I will argue that if semantic networks are to be used as a representation for storing human verbal knowledge, then they must include mechanisms for representing propositions without commitment to asserting their truth or belief. Also they must be able to represent various types of intensional objects without commitment to their existence in the external world, their external distinctness, or their completeness in covering all of the objects which are presumed to exist. I will discuss the problems of representing restrictive relative clauses and argue that a commonly used "solution" is inadequate. I will also demonstrate the inadequacy of certain commonly used techniques which purport to handle quantificational information in semantic networks. Three adequate mechanisms will be presented, one of which to my knowledge has not previously been used in semantic nets. I will discuss

several different possible uses of links and some of the different types of nodes and links which are required in a semantic network if it is to serve as a medium for representing knowledge.

The emphasis of the paper will be on problems, possible solution techniques, and necessary characteristics of solutions, with particular emphasis on pointing out non-solutions. No attempt will be made to formulate a complete specification of an adequate semantic network notation. Rather, the discussion will be oriented towards requirements for an adequate notation and the kind of explicit understanding of what one intends his notations to mean that are required to investigate such questions.

II What is Semantics

First we must come to grips with the term "semantics". What do semantic networks have to do with semantics? What is semantics anyway? There is a great deal of misunderstanding on this point among computational linguists and psychologists. There are people who maintain that there is no distinction between syntax and semantics, and there are others who lump the entire inference and "thought" component of an AI system under the label "semantics". Moreover, the philosophers, linguists, and programming language theorists

have notions of semantics which are distinct from each other and from many of the notions of computational linguists and psychologists.

What I will present first is my view of the way that the term "semantics" has come to be associated with so many different kinds of things, and the basic unity that I think it is all about. I will attempt to show that the source of many confusing claims such as "there is no difference between syntax and semantics" arise from a limited view of the total role of semantics in language.

1. The Philosopher and the Linguist

In my account of semantics, I will use some caricatured stereotypes to represent different points of view which have been expressed in the literature or seem to be implied. I will not attempt to tie specific persons to particular points of view since I may thereby make the error of misinterpreting some author. Instead, I will simply set up the stereotype as a possible point of view which someone might take, and proceed from there.

First, let me set up two caricatures which I will call the Linguist and the Philosopher, without thereby asserting that all linguists fall into the first category or philosophers in the second. However, both represent strong traditions in their respective fields. The Linguist has the

following view of semantics in linguistics: he is interested in characterizing the fact that the same sentence can sometimes mean different things, and some sentences mean nothing at all. He would like to find some notation in which to express the different things which a sentence can mean and some procedure for determining whether a sentence is "anomalous" (i.e., has no meanings). The Philosopher on the other hand is concerned with specifying the meaning of a formal notation rather than a natural language. (Again, this is not true of all philosophers -- just our caricature.) His notation is already unambiguous. What he is concerned with is determining when an expression in the notation is a "true" proposition (in some appropriate formal sense of truth) and when it is false. (Related questions are when it can be said to be necessarily true or necessarily false or logically true or logically false, etc.) Meaning for the Philosopher is not defined in terms of some other notation in which to represent different possible interpretations of a sentence, but he is interested in the conditions for truth of an already formal representation.

Clearly, these caricatured points of view are both parts of a larger view of the semantic interpretation of natural language. The Linguist is concerned with the translation of natural languages into formal representations of their meanings, while the Philosopher is interested in the meanings of such representations. One cannot really

have a complete semantic specification of a natural language unless both of these tasks have been accomplished. However, I will go further and point out that there is a consideration which the philosophers have not yet covered and which must be included in order to provide a complete semantic specification.

2. Procedural Semantics

While the types of semantic theories that have been formulated by logicians and philosophers do a reasonable job of specifying the semantics of complex constructions involving quantification and combination of predicates with operators of conjunction and negation, they fall down on the specification of the semantics of the basic "atomic" propositions consisting of a predicate and specifications of its arguments -- for example, the specification of the meanings of elementary statements such as "snow is white" or "Socrates is mortal". In most accounts, these are presumed to have "truth conditions" which determine those possible worlds in which they are true and those in which they are false, but how does one specify those truth conditions? In order for an intelligent entity to know the meaning of such sentences it must be the case that it has stored somehow an effective set of criteria for deciding in a given possible world whether such a sentence is true or false. Thus, it is not sufficient merely to say that the meaning of a sentence

is a set of truth conditions - one must be able to specify the truth conditions for particular sentences. Most philosophers have not faced this issue for atomic sentences such as "snow is white."

Elsewhere I have argued [Woods, 1967, 1973a] that a specification of truth conditions can be made by means of a procedure or function which assigns truth values to propositions in particular possible worlds. Such procedures for determining truth or falsity are the basis for what I have called "procedural semantics", (although this interpretation of the term may differ slightly from that which is intended by other people who have since used it). This notion has served as the basis of several computer question-answering systems [Woods, Kaplan and Nash-Webber, 1972; Woods, 1973b; Winograd, 1972].

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

inference techniques. Some of these issues are discussed more fully in Woods [1973a].

3. Semantic Specification of Natural Language

You now have the basics of my case for a broader view of the role of semantics in natural language. The outline of the picture goes like this:

There must be a notation for representing the meanings of sentences inside the brain (of humans or other intellects) that is not merely a direct encoding of the English word sequence. This must be so, since (among other reasons) what we understand by sentences usually includes the disambiguation of certain syntactic and semantic ambiguities present in the sentence itself.

The linguist is largely concerned with the process for getting from the external sentence to this internal representation (a process referred to as semantic interpretation). The philosopher is concerned with the rules of correspondence between expressions in such notations and truth and falsity (or correctness of assertion) in the real or in hypothetical worlds. However, philosophers have generally stopped short of trying to actually specify the truth conditions of the basic atomic propositions in their systems, dealing mainly with the specification of the meanings of complex expressions in

terms of the meanings of elementary ones. Researchers in artificial intelligence are faced with the need to specify the semantics of elementary propositions as well as complex ones and are moreover required to put to the test the assembly of the entire system into a working total - including the interface to syntax and the subsequent inference and "thought" processes. Thus, the researcher in artificial intelligence must take a more global view of the semantics of language than either the linguist or the philosopher has taken in the past. The same, I think, is true of psychologists.

4. Misconceptions about Semantics

There are two misconceptions of what semantics is about (or at least misuses of the term) that are rather widely circulated among computational linguists and which arise I think from a limited view of the role of semantics in language. They arise from traditional uses of the term but through specialized application eventually lose sight of what semantics is really about. According to my dictionary, semantics is "the scientific study of the relations between signs or symbols and what they denote or mean." This is the traditional use of the term and represents the common thread which links the different concerns discussed previously. Notice that the term does not refer to the things denoted or the meanings, but to the relations between these things and

the linguistic expressions which denote them.

One common misuse of the term "semantics" in the fields of computational linguistics and artificial intelligence is to extend the coverage of the term not only to this relation between linguistic form and meaning, but to all of the retrieval and inference capabilities of the system. This misuse arises since for many tasks in language processing, the use of semantic information necessarily involves not only the determination of the object denoted, but also some inference about that object. In absence of a good name for this further inference process, terms such as "semantic inferences" have come to be used for the entire process. It is easy then to start incorrectly referring to the entire thought process as "semantics." One may properly use the term "semantic inferences" to refer to inferences that cross the boundary between symbol and referent, keeping in mind, however, that this does not imply that all steps of the process are "semantic."

At the opposite extreme, there are those who deny any difference in principle between syntax and semantics and claim that the distinction is arbitrary. Again, the misconception arises from a limited view of the role of semantics. When semantics is used to select among different possible parsings of a sentence by using selectional restrictions on so-called semantic features of words, there

is little difference between the techniques usually used and those used for checking syntactic features. In another paper [Woods, 1973a] I make the case that such techniques are merely approximations of the types of inferences that are really required, and that in general, semantic selectional restrictions need to determine the referent of a phrase and then make inferences about that referent (i.e., they involve semantic inferences as I defined the term above). However, the approximate technique usually used requires no special mechanism beyond what already exists in the syntax specification, and when taken as the paradigm for "semantic inferences" can lead to the false conclusion that semantics is no different from syntax. Likewise, if the representation constructed by a parser purports to be a semantic representation, with no intervening purely syntactic representation, then one might argue that the techniques used to produce it are syntactic techniques and therefore there is nothing left to be semantics.

As we have pointed out, however, a semantic specification requires more than the transformation of the input sentence into a "semantic" representation. The meanings of these representations must be specified also. Recall that semantics refers to the correspondence between linguistic expressions and the things that they denote or mean. Thus, although it may be difficult to isolate exactly what part of a system is semantics, any system which

understands sentences and carries out appropriate actions in response to them is somehow completing this connection. For systems which do not extend beyond the production of a so-called semantic representation, there may or may not be a semantic component included, and the justification for calling something semantic may be lost. Again, if one takes the production of such "semantic" representations as the paradigm case for what semantics is, one is misunderstanding the meaning of the term.

5. Semantics of Programming Languages

Before proceeding it is probably worth pointing out that the use of the term semantics by programming language theorists has been much closer to the tradition of the logicians and the philosophers and less confused than in computational linguistics. Programming language theory is frequently used as a paradigm for natural language semantics. However, programming languages do not have many of the features that natural languages do and the mechanisms developed there are not sufficient for modeling the semantics of natural language without considerable stretching.

The programming language theorists do have one advantage over the philosophers and linguists in that their semantic specifications stand on firmer ground since they

are defined in terms of the procedures that the machine is to carry out. It is this same advantage which the notion of procedural semantics and artificial intelligence brings to the specification of the semantics of natural language. Although in ordinary natural language not every sentence is overtly dealing with procedures to be executed, it is possible nevertheless to use the notion of procedures as a means of specifying the truth conditions of declarative statements as well as the intended meaning of questions and commands. One thus picks up the semantic chain from the philosophers at the level of truth conditions and completes it to the level of formal specifications of procedures. These can in turn be characterized by their operations on real machines and can be thereby anchored to physics. (Notice that the notion of procedure shares with the notion of meaning that elusive quality of being impossible to present except by means of alternative representations. The procedure itself is something abstract which is instantiated whenever someone carries out the procedure, but otherwise, all one has when it is not being executed is some representation of it.)

III. Semantics and Semantic Networks

Having established a framework for understanding what we mean by semantics, let us now proceed to see how semantic

networks fit into the picture. Semantic networks presumably are candidates for the role of internal semantic representation -- i.e., the notation used to store knowledge inside the head. Their competitors for this role are formal logics such as the predicate calculus, and various representations such as Lakoff-type deep structures, and Fillmore-type case representations. (The case representations shade off almost imperceptibly into certain possible semantic network representations and hence it is probably not fruitful to draw any clear distinction.) The major characteristic of the semantic networks that distinguishes them from other candidates is the characteristic notion of a link or pointer which connects individual facts into a total structure.

A semantic network attempts to combine in a single mechanism the ability not only to store factual knowledge but also to model the associative connections exhibited by humans which make certain items of information accessible from certain others. It is possible presumably to model these two aspects with two separate mechanisms such as, for example, a list of the facts expressed in the predicate calculus or some such representation, together with an index of associative connections which link facts together. Semantic network representations attempt instead to produce a single representation which by virtue of the way in which it represents facts (i.e., by assemblies of pointers to

other facts) automatically provides the appropriate associative connections. One should keep in mind that the assumption that such a representation is possible is merely an item of faith, an unproven hypothesis used as the basis of the methodology. It is entirely conceivable that no such single representation is possible.

1. Requirements for a Semantic Representation

When one tries to devise a notation or a language for semantic representation, he is seeking a representation which will precisely, formally, and unambiguously represent any particular interpretation that a human listener may place on a sentence. We will refer to this as "logical adequacy" of a semantic representation. There are two other requirements of a good semantic representation beyond the requirement of logical adequacy. One is that there must be an algorithm or procedure for translating the original sentence into this representation and the other is that there must be algorithms which can make use of this representation for the subsequent inferences and deductions that the human or machine must perform on them. Thus, one is seeking a representation which facilitates translation and subsequent intelligent processing, in addition to providing a notation for expressing any particular interpretation of a sentence.

2. The Canonical Form Myth

Before continuing, let me mention one thing which semantic networks should not be expected to do. That is to provide a "canonical form" in which all paraphrases of a given proposition are reduced to a single standard (or canonical) form. It is true that humans seem to reduce input sentences into some different internal form that does not preserve all of the information about the form in which the sentence was received (e.g., whether it was in the active or the passive). A canonical form, however, requires a great deal more than this. A canonical form requires that every expression equivalent to a given one can be reduced to a single form by means of an effective procedure, so that tests of equivalence between descriptions can be reduced to the testing of identity of canonical form. I will make two points. The first is that it is unlikely that there could be a canonical form for English, and the second is that for independent reasons, in order to duplicate human behavior in paraphrasing, one would still need all of the inferential machinery that canonical forms attempt to avoid.

Consider first the motivation for wanting a canonical form. Given a system of expressions in some notation (in this case English, or more specifically an internal semantic representation of English) and given a set of equivalence preserving transformations (such as paraphrasing or logical

equivalence transformations) which map one expression into an equivalent expression, two expressions are said to be equivalent if one can be transformed into the other by some sequence of these equivalence transformations. If one wanted to determine if two expressions e1 and e2 were equivalent, he would expect to have to search for a sequence of transformations that would produce one from the other -- a search which could be non-deterministic and expensive to carry out. A canonical form for the system is a computable function c which transforms any expression e into a unique equivalent expression c(e) such that for any two expressions e1 and e2, e1 is equivalent to e2 if and only if c(e1) is equal to c(e2). With such a function, one can avoid the combinatoric search for an equivalence chain connecting the two expressions and merely compute the corresponding canonical forms and compare them for identity. Thus a canonical form provides an improvement in efficiency over having to search for an equivalence chain for each individual case (assuming that the function c is efficiently computable).

A canonical form function is, however, a very special function, and it is not necessarily the case for a given system of expressions and equivalence transformations that there is such a function. It can be shown for certain formal systems such as the word problem for semigroups [Davis, 1958] that there can be no computable canonical form

function with the above properties. That is, in order to determine the equivalence of a particular pair of expressions e1 and e2 it may be necessary to actually search for a chain of equivalence transformations that connects these two particular expressions rather than performing separate transformations c(e1) and c(e2) (both of which know exactly where to stop) and then compare these resulting expressions for identity. If this can be the case for formal systems as simple as semigroups, it would be foolhardy to lightly assume that there is a canonical form for something as complex as English paraphrasing.

Now, for the second point. Quite aside from the possibility of having a canonical form function for English, I will attempt to argue that one still needs to be able to search for individual chains of inference between pairs of expressions e1 and e2 and thus the principal motivation for wanting a canonical form is superfluous. The point is that in most cases where one is interested in some paraphrase behavior, the paraphrase desired is not one of full logical equivalence, but only of implication in one direction. For example, one is interested in whether the truth of some expression e1 is implied by some stored expression e2. If one had a canonical form function, then one could store only canonical forms in the data base and ask simply whether c(e1) is stored in the data base without having to apply any equivalence transformations in the process. However, this

is just a special case. It is rather unlikely that what we have in the data base is an expression exactly logically equivalent to e1 (i.e., some e2 such that e2 implies e1 and e1 implies e2). Rather, what we expect in the typical case is that we will find some e2 that implies e1 but not vice versa. For this case, we must be able to find an inference chain as part of our retrieval process. Given that we must devise an appropriate inferential retrieval process for dealing with this case (which is the more common) the special cases of full equivalence will fall out as a consequence and thus the canonical form mechanism for handling the full equivalence case gives no improvement in performance and is unnecessary.

There is still benefit from "partially canonicalizing" the stored knowledge (the term is reminiscent of the concept of being just a little bit pregnant). This is useful to avoid storing multiple equivalent representations of the same fact. However, there is little motivation for making sure that this form does in fact reduce all equivalent expressions to the same form (and as I said before, there is every reason to believe that this may be impossible).

Another argument against the expectation of a canonical form solution to the equivalence problem comes from the following situation. Consider the kinship relations program of Lindsay [1963]. The basic domain of discourse of the system is family relationships such as mother, father, brother, sister, etc. The data structure chosen is a logically minimal representation of a family unit consisting of a male and female parent and some number of offspring. Concepts such as aunt, uncle, and brother-in-law are not represented explicitly in the structure but are rather implicit in the structure, and questions about unclehood are answered by checking brothers of the father and brothers of the mother. However, what does such a system do when it encounters the input "Harry is John's Uncle"? It doesn't know whether to assign Harry as a sibling of John's father or his mother. Lindsay had no good solution for this problem other than the suggestion to somehow make both entries and connect them together with some kind of a connection which indicates that one of them is wrong. It seems that for handling "vague" predicates such as uncle, i.e., predicates which are not specific with respect to some of the details of an underlying representation, we must make provision for storing such predicates directly (i.e., in terms of a concept of uncle in this case), even though the concept may be defined in terms of more "basic" relationships (ignoring here the issue that there may be no

objective criterion for selecting any particular set of relationships as basic).

If we hope to be able to store information at the level of detail that it may be presented to us in English, then we are compelled to surrender the assumptions of logical minimality in our internal representation and provide for storing such redundant concepts as "uncle" directly. However, we would not like to have to store all such facts redundantly. That is, given a Lindsay-type data base of family units, we would not want to be compelled to explicitly store all of the instances of unclehood that could be inferred from the basic family units. If we were to carry such a program to its logical conclusion we would have to explicitly store all of the possible inferable relations, a practical impossibility since in many cases the number of such inferables is effectively infinite. Hence the internal structure that we desire must have some instances of unclehood stored directly and others left to be deduced from more basic family relationships, thus demolishing any hope of a canonical form representation.

3. Semantics of Semantic Network Notations

When I create a node in a network or when I establish a link of some type between two nodes, I am building up a representation of something in a notation. The question

that I will be concerned with in the remainder of this paper is what do I mean by this representation. For example, if I create a node and establish two links from it, one labeled SUPERC and pointing to the "concept" TELEPHONE and another labeled MOD and pointing to the "concept" BLACK, what do I mean this node to represent? Do I intend it to stand for the "concept" of a black telephone, or perhaps I mean it to assert a relationship between the concepts of telephone and blackness -- i.e., that telephones are black (all telephones?, some telephones?). When one devises a semantic network notation, it is necessary not only to specify the types of nodes and links that can be used and the rules for their possible combinations (the syntax of the network notation) but also to specify the import of the various types of links and structures -- what is meant by them (the semantics of the network notation).

4. Intensions and Extensions

To begin, I would like to raise the distinction between intension and extension, a distinction that has been variously referred to as the difference between sense and reference, meaning and denotation, and various other pairs of terms. Basically a predicate such as the English word "red" has associated with it two possible conceptual things which could be related to its meaning in the intuitive sense. One of these is the set of all red things -- this is

called the extension of the predicate. The other concept is an abstract entity which in some sense characterizes what it means to be red, it is the notion of redness which may or may not be true of a given object; this is called the intension of the predicate. In many philosophical theories the intension of a predicate is identified with an abstract function which applies to possible worlds and assigns to any such world a set of extensional objects (e.g., the intension of "red" would assign to each possible world a set of red things). In such a theory, when one wants to refer to the concept of redness, what is denoted is this abstract function.

5. The Need for Intensional Representation

The following quote from Quine [1961] relating an example of Frege should illustrate the kind of thing that I am trying to distinguish as an internal intensional entity:

> "The phrase 'Evening Star' names a certain large physical object of spherical form, which is hurtling through space some scores of millions of miles from here. The phrase 'Morning Star' names the same thing, as was probably first established by some observant Babylonian. But the two phrases cannot be regarded as having the same meaning; otherwise that Babylonian could have dispensed with his observations and contented himself with reflecting on the meanings of his words. The meanings, then, being different from one another, must be other than the named object, which is one and the same in both cases." [Quine, 1961, p 9].

In the appropriate internal representation, there must be two mental entities (concepts, nodes, or whatever) corresponding to the two different intensions, morning star and evening star. There is then an assertion about these two intensional entities that they denote one and the same external object (extension).

In artificial intelligence applications and psychology, it is not sufficient for these intensions to be abstract entities such as possibly infinite sets, but rather they must have some finite representation inside the head as it were, or in our case in the internal semantic representation.

6. Attributes and "Values"

Much of the structure of semantic networks is based on, or at least similar to, the notion of attribute and value which has become a standard concept in a variety of computer science applications and was the basis of Raphael's SIR program [Raphael, 1964] -- perhaps the earliest forerunner of today's semantic networks. Facts about an object can frequently be stored on a "property list" of the object by specifying such attribute-value pairs as HEIGHT : 6 FEET, HAIRCOLOR : BROWN, OCCUPATION : SCIENTIST, etc. (Such lists are provided, for example, for all atoms in the LISP programming language.) One way of thinking of these pairs is

that the attribute name (i.e., the first element of the pair) is the name of a "link" or "pointer" which points to the "value" of the attribute (i.e., the second element of the pair). Such a description of a person named John might be laid out graphically as:

JOHN

HEIGHT	6 FEET
HAIRCOLOR	BROWN
OCCUPATION	SCIENTIST

Now it may seem the case that the intuitive examples that I just gave are all that it takes to explain what is meant by the notion of attribute-value pair and that the use of such notations can now be used as part of a semantic network notation without further explanation. I will try to make the case that this is not so, and thereby give a simple introduction to the kinds of things I mean when I say that the semantics of the network notation need to be specified.

The above examples seem to imply that the thing which occurs as the second element of an attribute-value pair is the name or at least some unique handle on the value of that attribute. However, what will I do with an input sentence "John's height is greater than 6 feet". Most people would not hesitate to construct a representation such as:

JOHN

HEIGHT (GREATERTHAN 6 FEET)

Notice, however, that our interpretation of what our network notations mean has just taken a great leap. No longer is the second element of the attribute-value pair a name or a pointer to a value, but rather it is a predicate which is asserted to be true of the value. One can think of the names such as 6 FEET and BROWN in the previous examples as special cases of identity predicates which are abbreviated for the sake of conciseness and thereby consider the thing at the end of the pointer to be always a predicate rather than a name. Thus, there are at least two possible interpretations of the meaning of the name at the end of the link -- either as the name of the value or as a predicate which must be true of the value. The former will not handle the (GREATERTHAN 6 FEET) example, while the latter will.

Let us consider now another example -- "John's height is greater than Sue's". We now have a new set of problems. We can still think of a link named HEIGHT pointing from JOHN to a predicate whose interpretation is "greater than Sue's height", but what does the reference to Sue's height inside this predicate have to do with the way that we represented John's height? In a functional form we would simply represent this as HEIGHT(JOHN) > HEIGHT(SUE), or in LISP type "Cambridge Polish" notation,

(GREATER (HEIGHT JOHN)(HEIGHT SUE))

but that is departing completely from the notion of attribute-value links. There is another possible interpretation of the thing at the end of the HEIGHT link which would be capable of dealing with this type of situation. That is, the HEIGHT link can point from JOHN to a node which represents the intensional object "John's height". In a similar way, we can have a link named HEIGHT from SUE to a node which represents "Sue's height" and then we can establish a relation GREATER between these two intensional nodes. (Notice that even if the heights were the same, the two intensional objects would be different, just as in the morning star/evening star example.) This requires a major reinterpretation of the semantics of our notation and a new set of conventions for how we set up networks. We must now introduce a new intensional node at the end of each attribute link and then establish predicates as facts that are true about such intensional objects. It also raises for us a need to somewhere indicate about this new node that it was created to represent the concept of John's height, and that the additional information that it is greater than Sue's height is not one of its defining properties but rather a separate assertion about the node. Thus, a distinction between defining and asserted properties of the node become important here. In my conception of semantic networks I have used the concept of an EGO link to indicate for the benefit of the human researcher and

eventually for the benefit of the system itself what a given node is created to stand for. Thus the EGO's of these two nodes are John's height and Sue's height respectively. The EGO link represents the intensional identity of the node.

7. Links and Predication

In addition to considering what is at the end of a link, we must also consider what the link itself means. The examples above suggest that an attribute link named Z from node X to Y is equivalent to the English sentence "the Z of X is Y" or functionally Z(X)=Y or (in the case where Y is a predicate) Y(Z(X)), (read Y of Z of X). Many people, however, have used the same mechanism and notation (and even called it attribute-value pairs) to represent arbitrary English verbs by storing a sentence such as "John hit Mary" as a link named HIT from the node for John to the node for Mary, as in the structure:

```
JOHN

        HIT     MARY
```

and perhaps placing an inverse link under Mary:

```
MARY

        HIT*    JOHN
```

If we do this, then suddenly the semantics of our notation has changed again. No longer do the link names stand for attributes of the node, but rather arbitrary relations

between the node and other nodes. If we are to mix the two notations together as in:

JCHN

HEIGHT	6 FEET
HIT	MARY

then we need either to provide somewhere an indication that these two links are of different types and therefore must be treated differently by the procedures which make inferences in the net, or else we need to find a unifying interpretation such as considering that the "attribute" HEIGHT is now really an abbreviation of the relation "height of equals" which holds between JOHN and (the node?) 6 FEET. It is not sufficient to leave it to the intuition of the reader, we must know how the machine will know to treat the two arcs correctly.

If we use Church's lambda notation, which provides a convenient notation for naming predicates and functions constructed out of combinations or variations of other functions (this is used, for example, as the basic function specification notation in the LISP programming language), we could define the meaning of the height link as the relation (LAMBDA (X Y) (EQUAL (HEIGHT X) Y)). By this we mean the predicate of two arguments X and Y which is true when and only when the height of X is equal to Y. Thus, a possible unifying interpretation of the notation is that the link is

always the name of a relation between the node being described and the node pointed to, (providing that we reinterpret what we meant by the original link named HEIGHT). Whatever we do, we clearly need some mechanism for establishing relations between nodes as facts (e.g., to establish the above GREATER relation between the nodes for John's height and Sue's height).

8. Relations of More than Two Arguments

In the example just presented, we have used a link to assert a relation between two objects in the network corresponding to the proposition that John hit Mary. Such a method of handling assertions has a number of disadvantages, perhaps the simplest of which is that it is constrained to handling binary relations. If we have a predicate such as the English preposition "between" (i.e., (LAMBDA (X Y Z) (Y is between X and Z))), then we must invent some new kind of structure for expressing such facts. A typical, but not very satisfying, notation that one might find in a semantic network of the type which uses links for relations is something like:

```
        Y
LOCATION    (BETWEEN X Z)
```

usually without further specification of the semantics of the notation or what kind of thing the structure (BETWEEN X

Z) is. For example, is it the name of a place? In some implementations it would be exactly that, in spite of the fact that an underlying model in which there is only one place between any given pair of places is an inadequate model of the world we live in. Another possible interpretation is that it denotes the range of places between the two endpoints (this interpretation requires another interpretation of what the LOCATION link means -- the thing at the end is no longer a name of a place but rather a set of places, and the LOCATION link must be considered to be implicitly existentially quantified in order to be interpreted as asserting that the location is actually one of those places and not all).

Given the notion which we introduced previously that interprets the thing at the end of the link as a predicate which must be true of the location, we have perhaps the best interpretation -- we can interpret the expression (BETWEEN X Z) at the end of the link as being an abbreviation for the predicate (LAMBDA (U) (BETWEEN X U Z)), i.e., a one place predicate whose variable is U and whose values of X and Z are fixed to whatever X and Z are.

Although this representation of the three-place predicate "between" (when supplied with an appropriate interpretation of what it means) seems plausible, and I see no major objections to it on the grounds of logical

inadequacy, one is left with the suspicion that there may be some predicates of more than two places which don't have such an intuitively satisfying decomposition into links connecting only two objects at a time. For example, I had to introduce the concept of location as the name of the link from Y to the special object (BETWEEN X Z). In this case, I was able to find a preexisting English concept which made the creation of this link plausible, but is this always the case? The account would have been much less satisfying if all I could have produced was something like:

X

BETWEEN1 (BETWEEN2 Y Z)

with an explication of its semantics that (BETWEEN2 Y Z) was merely some special kind of entity which when linked to X by a BETWEEN1 link represented the proposition (BETWEEN X Y Z). It may be the case that all predicates in English with more than two arguments have a natural binary decomposition. The basic subject-predicate distinction which seems to be made by our language gives some slight evidence for this. It seems to me, however, that finding a natural binary decomposition for sentences such as "John sold Mary a book" (or any of Schank's various TRANS operations) is unlikely.

9. Case Representations in Semantic Networks

Another type of representation is becoming popular in semantic networks and handles the problem of relations of more than one argument very nicely. This representation is based on the notion of case introduced by Fillmore [1968]. Fillmore advocates a unifying treatment of the inflected cases of nominals in Latin and other highly inflected languages and the prepositions and positional clues to role that occur in English and other largely noninflected languages. A case as Fillmore uses the term is the name of a particular role that a noun phrase or other participant takes in the state or activity expressed by the verb of a sentence. In the case of the sentence "John sold Mary a book" we can say that John is the agent of the action, Mary is the recipient or beneficiary of the action, and the book is the object or patient of the action (where I have taken arbitrary but typical names for the case roles involved for the sake of illustration). When such a notation is applied to semantic network representations, a major restructuring of the network and what it means to be a link takes place. Instead of the assertion of a fact being carried by a link between two nodes, the asserted fact is itself a node. Our structure might look something like:

SELL

AGT	JOHN
RECIP	MARY
PAT	BOOK

(ignoring for the moment what has happened to turn "a book" into BOOK or for that matter what we mean by JOHN and MARY -- we will get into that later). The notation as I have written it requires a lot of explanation, which is unfortunately not usually spelled out in the presentation of a semantic network notation. In our previous examples, the first item (holding the position where we have placed SELL above) has been the unique name or "handle" on a node and the remaining link-value pairs have been predicates that are true of this node. In the case above, which I have written that way because one is likely to find equivalent representations in the literature, we are clearly not defining characteristics of the general verb "sell", but rather setting up a description of a particular instance of selling. Thus, to be consistent with our earlier format for representing a node we should more properly represent it as something like:

S13472

VERB	SELL
AGT	JOHN
RECIP	MARY
PAT	BOOK

where S13472 is some unique internal handle on the node representing this instance of selling, and SELL is now the internal handle on the concept of selling. (I have gone through this two-stage presentation in order to emphasize that the relationship between the node S13472 and the

concept of selling is not essentially different at this level from the relationship it has to the other nodes which fill the cases.)

10. Assertional and Structural Links

Clearly the case structure representation in a semantic network places a new interpretation on the nodes and arcs in the net. We still seem to have the same types of nodes that we had before for JOHN, MARY, etc., but we have a new type of node for nodes such as S13472 which represent assertions or facts. Moreover, the import of the links from this new type of node is different from that of our other links. Whereas the links which we discussed before are assertional, i.e., their mere presence in the network represents an assertion about the two nodes that they connect, these new link names VERB, AGT, RECIP, PAT are merely setting up parts of the proposition represented by node S13472, and no single link has any assertional import by itself; rather these links are definitional or structural in that they constitute the definition of what node S13472 means.

Now you may argue that these links are really the same as the others; i.e., they correspond to the assertion that the agent of S13472 is JOHN and that S13472 is an instance of selling, etc., just like the "hit" link between John and Mary in our previous example. In our previous example,

however, the nodes for John and for Mary had some a priori meanings independent of the assertion of hitting that we were trying to establish between them. In this case, S13472 has no meaning other than that which we establish by virtue of the structural links which it has to other nodes. That is, if we were to ask for the ego of the node S13472, we would get back something like "I am an instance of John selling a book to Mary" or "I am an instance of selling whose agent is John, whose recipient is Mary and whose patient is a book." If we were to ask for the ego of JOHN, we would get something like "I am the guy who works in the third office down the hall, whose name is John Smith, etc." The fact which I am trying to assert with the "hit" link is not part of the ego of JOHN or else I would not be making a new assertion.

This difference between assertional and structural links is rather difficult for some people to understand, and is often confused in various semantic network representations. It is part of the problem that we cited earlier in trying to determine whether a structure such as:

N12368

SUPERC	TELEPHONE
MOD	BLACK

is to be interpreted as an intensional representation of a black telephone or an assertion that telephones are black.

If it is to be interpreted as an intensional representation of the concept of a black telephone, then both of these links are structural or definitional. If on the other hand, it is to be interpreted as asserting that telephones are black then the first link is structural while the second is assertional. (The distinction between structural and assertional links does not take care of this example entirely since we still have to worry about how the assertional link gets its quantificational import for this interpretation, but we will discuss this problem later.)

The above discussion barely suffices to introduce the distinction between structural and assertional links, and certainly doesn't make the distinction totally clear. Moreover, before we are through, we may have cause to repudiate the assumption that the links involved in our non-case representation should be considered to have assertional import. Perhaps the best way to get deeper into the problems of different types of links with different imports and the representation of intensional entities is to consider further some specific problems in knowledge representation.

IV. Problems in Knowledge Representation

In previous sections I hope that I have made the point

that the same semantic network notations could be used by different people (or even by the same person at different times for different examples) to mean different things, and therefore one must be specific in presenting a semantic network notation to make clear what he means by the notations which he uses (i.e., the semantics of the notation). In the remainder of this paper, I would like to discuss two difficult problems of knowledge representation and use the discussion to illustrate several additional possible uses of links and some of the different types of nodes and links which are required in a semantic network if it is to serve as a medium for representing human verbal knowledge. The specific problems which I will consider are the representation of restrictive relative clauses and the representation of quantified information.

1. Relative Clauses

In attaching modifiers to nodes in a network to provide an intensional description for a restricted class, one often requires restrictions which do not happen to exist in the language as single-word modifiers but have to be constructed out of more primitive elements. The relative clause mechanism permits this. Anything that can be said as a proposition can be used as a relative clause by leaving some one of its argument slots unfilled and using it as a modifier. (We will be concerned here only with restrictive

relative clauses and not those which are just parenthetical comments about an already determined object.) Let me begin my discussion of relative clauses by dispensing with one inadequate treatment.

1.1. The Shared Subpart Fallacy

A mechanism which occasionally surfaces as a claimed technique for dealing with relative clauses is to simply take the two propositions involved, the main clause and the relative clause, and represent the two separately as if they were independent propositions. In such a representation, the sentence "The dog that bit the man had rabies" would look something like that in Fig. 1.

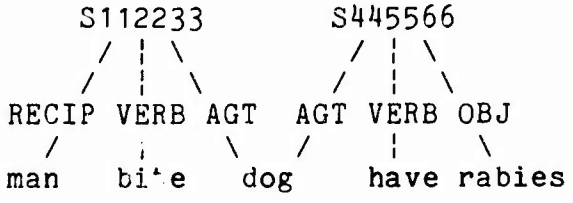


Fig. 1. A shared subpart representation

The point of interest here is not the names of the links (for which I make no claims) nor the type of representation (case oriented, deep conceptual, or whatever), but simply the fact that the only relationship between the two propositions is that they share the same node for dog. There are a number of problems with this representation: First, since there is no other relationship between the two sentences except sharing of a node (which is a symmetric relationship) there is no indication of which is the main clause and which is the relative clause. That is, we would get the same internal representation for the sentence "The dog that had rabies bit the man."

Another difficulty is that there is nothing to indicate that the two sentences go together at all in a relative clause relationship. It is possible that on two different occasions we were told about this dog. On one occasion that he had rabies and on another that he bit a man. Then the presence of the two propositions in our data base both sharing the same node for dog would give us a structure identical to that for the example sentence. Now there is a subtle confusion which can happen at this point which I would like to try to clarify. You may say to me, "So what is the problem. Suppose I tell you about this dog and suppose I have told you the two facts at different times, then it is still true that the dog that bit the man has rabies." How do I answer such an argument? On the face of it

it seems true. Yet I maintain that the argument is fallacious and that it results from too shallow a treatment of the issues. The crux of the matter I think rests in the notion of which dog we are talking about. Unfortunately, this issue is one that gets omitted from almost all such discussions of semantic networks. If the two facts were told to me at different times, how did I know that they were about the same dog? (Without further explication of the semantics of the network notation, it is not even clear that we are talking about a particular dog and not about dogs in general.) It is exactly in order to relate the second fact to the first that we need the relative clause mechanism. In the next section we will consider the problem in more detail.

1.2. The Transient-Process "Account"

Quillian [personal communication] once made the observation that a portion of what was in an input sentence was essentially stage directions used to enable the understanding process to identify an appropriate internal concept or node and the rest of the utterance was to be interpreted as new information to be added somehow to the network (and similar observations have been made by others). This gives an attractive account of the relative clause problem above. We interpr the relative clause not as something to be added to t. network at all, but rath.r as a

description to be used by the understander to determine which dog is in question. After this, we can forget about the relative clause (it has served its usefulness) and simply add the new information to the network. We might call this the "transient-process account." Under this account, if I was told about a dog that bit a man and later told that the dog that bit the man had rabies, then I would simply use the relative clause to find the internal concept for the dog that bit the man, and then add the new information that the dog had rabies. What's wrong with that account? Doesn't that explain everything?

Well, no. First, it simply evades the issue of representing the meaning of the sentence, focusing instead on the resulting change in memory contents. It says essentially that the role of the relative clause is a temporary and transient one that exists only during the processing of the utterance and then goes away. But you say, "well, isn't that a plausible account, doesn't that take care of the problem nicely, who says you have to have a representation of the original sentence anyway?"

Let's start from the first question -- yes, it is a plausible account of the interpretation of <u>many</u> sentences, including this one in the context I just set up. And it may also be a correct description of what happens when humans process such sentences. But it doesn't take care of all

occurrences of relative clauses. What about a situation when I read this sentence out of context and I haven't heard about the dog before? Then my processing must be different. I must infer that there must be a dog that I don't know about, perhaps create a new node for it, and then assert about this new node that it has rabies. Clearly also I must associate with this new node that it is a dog and that it bit a man. But how do I keep these two different types of information separate -- the information which designates what I set the node up to stand for and that which the sentence asserted about it. We're back to the same problem. We need to distinguish the information that is in the relative clause from that in the main clause.

One possible way would be the use of an EGO link which points to a specification of what the node represents. Using such a link, when one creates the new node for the dog which bit the man, one would give the new node an EGO link which in essence says "I am the node which represents the dog that bit the man." When one then adds information to this node asserting additional facts about it, the original motivation for creating the node in the first place is not forgotten and the difference between the sentences "The dog that bit the man had rabies" and "the dog that had rabies bit the man" would lie in whether the facts about biting or about rabies were at the end of the EGO link. (There are a number of other questions which would require answers in

order to complete the specification of the use of EGO links for this purpose -- such as whether the propositions at the end of the EGO link are thereby made indirectly available as properties of this node or whether they are redundantly also included in the same status as the additional asserted properties which come later. We will not, however, go into these issues here.)

The above argument should have convinced you that the simple explanation of using relative clauses always only to identify preexisting nodes does not cover all of the cases. For certain sentences such as the above example, the object determined by the relative clause does not previously exist and something must be created in the semantic network which will continue to exist after the process is finished. This thing must have an internal representation which preserves the information that it is an object determined by a relative clause.

A second argument against the transient process account is that even for sentences where nothing needs to remain in memory after the process has completed (because the relative clause has been used to locate a preexisting node), something needs to be extracted from the input sentence which describes the node to be searched for. In our previous example something like the proposition "the dog bit the man" needs to be constructed in order to search for its

instances, and the process must know when it finds such an instance that it is the dog that is of interest and not the man. This specification of the node to be searched for is exactly the kind of thing which a semantic interpretation for the noun phrase "the dog that bit the man" should be. Thus, even when no permanent representation of the relative clause needs to remain after the understanding process has completed, something equivalent to it still needs to be constructed as part of the input to the search process. The transient process account does not eliminate the need for such a representation, and the issue of whether a complete representation of the entire sentence (including the relative clause) gets constructed and sent off to the understanding process as a unit or whether small pieces get created and sent off independently without ever being assembled into a complete representation is at this point a red herring. The necessary operations which are required for the search specification are sufficient to construct such a representation, and whether it is actually constructed or whether parts of it are merely executed for effect and then cast away is a totally separate question.

A third argument against the transient process account, which should have become apparent in the above discussion is that it is not an account at all, but merely a way of avoiding the problem. By claiming that the relative clause is handled during the transient process we have merely

pushed the problem of accounting for relative clauses off onto the person who attempts to characterize the understanding process. We have not accounted for it or solved it.

2. Representation of Complex Sentences

Let us return to the question of whether one needs a representation of the entire sentence as a whole or not. More specifically, does one need a representation of a proposition expressed about a node which itself has a propositional restriction, or can one effectively break this process up in such a way that propositions are always expressed about definite nodes. This is going to be a difficult question to answer because there is a sense in which even if the answer is the former, one can model it with a process which first constructs the relative clause restricted node and then calls it definite and represents the higher proposition with a pointer to this new node. The real question, then, is in what sense is this new node definite. Does it always refer to a single specific node like the dog in our above example, or is it more complicated than that? I will argue the latter.

3. Definite and Indefinite Entities

Consider the case which we hypothesized in which we had

to infer the existence of a heretofore unknown dog because we found no referent for "the dog that bit the man". This new node still has a certain definiteness to it. We can later refer to it again and add additional information, eventually fleshing it out to include its name, who owns it, etc. As such it is no different from any other node in the data base standing for a person, place, thing, etc. It got created when we first encountered the object denoted (or at least when we first recognized it and added it to our memory) and has subsequently gained additional information and may in the future gain additional information still. We know that it is a particular dog and not a class of dogs and many other things about it.

Consider, however, the question "Was the man bitten by a dog that had rabies?" Now, we have a description of an indefinite dog and moreover we have not asserted that it exists but merely questioned its existence. Now you may first try to weasel out of the problem by saying something like, "Well, what happens is that we look in our data base for dogs that have rabies in the same way that we would in the earlier examples, and finding no such dog, we answer the question in the negative." This is another example of pushing the problem off onto the understanding process, it doesn't solve it or account for it, it just avoids it (not to mention the assumption that the absence of information from the network implies its falsity).

Let us consider the process more closely. Unless our process were appropriately constructed (how?) it would not know the difference (at the time it was searching for the referent of the phrase) between this case and the case of an assertion about an unknown dog. Hence the process we described above would create a new node for a dog that has rabies unless we block it somehow. Merely asking whether the main clause is a question would not do it, since the sentence "Did the dog that bit the man have rabies" still must have the effect of creating a new definite node. (This is due to the effect of the presupposition of the definite singular determiner "the" that the object described must exist.) Nor is it really quite the effect of the indefinite article "a", since the sentence "a dog that had rabies bit the man" should still create a definite node for the dog. We could try conditions on questioned indefinites. Maybe that would work, but let me suggest that perhaps you don't want to block the creation of the new node at all but rather simply allow it to be a different type of entity, one whose existence in the real world is not presupposed by an intensional existence in the internal semantic network.

If we are to take this account of the hypothetical dog in our question, then we have made a major extension in our notion of structures in a semantic network and what they mean. Whereas previously we construed our nodes to correspond to real existing objects, now we have introduced

a new type of node which does not have this assumption. Either we now have two very different types of nodes (in which case we must have some explicit indicator or other mechanism in the notation to indicate the type of every node) or else we must impose a unifying interpretation. If we have two different types of nodes, then we still have the problem of telling the process which constructs the nodes which type of node to construct in our two examples.

One possible unifying interpretation is to interpret every node as an intensional description and assert an explicit predicate of existence for those nodes which are intended to correspond to real objects. In this case, we could either rely on an implicit assumption that intensional objects used as subjects of definite asserted sentences (such as "the dog that bit the man had rabies") must actually exist, or we could postulate an inferential process which draws such inferences and explicitly asserts existence for such entities.

Since the above account of the indefinite relative clause in our example requires such a major reinterpretation of the fundamental semantics of our network notations, one might be inclined to look for some other account that was less drastic. However, I will argue that such internal intensional entities are required in any case to deal with other problems in semantic representation. For example,

whenever a new definite node gets created, it may in fact stand for the same object as some other node that already exists, but the necessary information to establish the identity may only come later or not at all. This is a fundamental characteristic of the information that we must store in our nets. Consider again Frege's morning star / evening star example. Even such definite descriptions, then, are essentially intensional objects. (Notice as a consequence that one cannot make negative identity assertions simply on the basis of distinctness of internal semantic representations.)

Perhaps the strongest case for intensional nodes in semantic networks comes from referentially "opaque" verbs such as "need" and "want". When one asserts a sentence such as "I need a wrench", one does not thereby assert the existence of the object desired. However, one must include in the representation of this sentence some representation of the thing needed. For this interpretation, the object of the verb "need" should be an intensional description of the needed item. (It is also possible for the slot filler to be a node designating a particular entity rather than just a description, thus giving rise to an ambiguity of interpretation of the sentence. That is, is it a particular wrench that is needed, or will any wrench do?)

4. Consequences of Intensional Nodes

We conclude that there must be some nodes in the semantic network which correspond to descriptions of entities rather than entities themselves. Does that fix up the problem? Well, we have to do more than just make the assumption. We have to decide how to tell the two kinds of nodes apart, how we decide for particular sentences which type to create, and how to perform inferences on these nodes. If we have nodes which are intensional descriptions of entities, what does it mean to associate properties with the nodes or to assert facts about the nodes. We can't just rely on the arguments that we made when we were assuming that all of the nodes corresponded to definite external entities. We must see whether earlier interpretations of the meanings of links between nodes still hold true for this new expanded notion of node or whether they need modification or reinterpretation. In short we must start all over again from the beginning but this time with attention to the ability to deal with intensional descriptions.

Let me clarify further some of the kinds of things which we must be able to represent. Consider the sentence "Every boy loves his dog". Here we have an indefinite node for the dog involved which will not hold still. Linguistically it is marked definite (i.e., the dog that

belongs to the boy), but it is a variable definite object whose reference changes with the boy. There are also variable entities which are indefinite as in "Every boy needs a dog." Here we plunge into the really difficult and crucial problems in representing quantification. It is easy to create simple network structures that model the logical syllogisms by creating links from subsets to supersets, but the critical cases are those like the above. We need the notion of an intensional description for a variable entity.

To summarize, then, in designing a network to handle intensional entities, we need to provide for definite entities that are intended to correspond to particular entities in the real world, indefinite entities which do not necessarily have corresponding entities in the real world, and definite and indefinite variable entities which stand in some relation to other entities and whose instantiations will depend on the instantiations of those other entities.

5. Functions and Predicates

Another question about the interpretation of links and what we mean by them comes in the representation of information about functions and predicates. Functions and predicates have a characteristic that clearly sets them apart from the other types of entities which we have mentioned (with the possible exception of the variable

entity which depends on others) -- namely, they take arguments. Somewhere in the internal representation of an entity which is a function or a predicate there must be information about the arguments which the function or predicate takes, what kinds of entities can fill those arguments, and how the value of the function or the truth of the predicate is determined or constrained by the values of the arguments. There is a difference between representing the possible entities that can serve as arguments for a predicate and expressing the assertion of the predicate for particular values or classes of values of those arguments. Unfortunately this distinction is often confused in talking about semantic networks. That is, it is all too easy to use the notation:

```
LOVE
        AGT    HUMAN
        RECIP  HUMAN
```

to express constraints on the possible fillers for the arguments of the predicate and to use the same link names in a notation such as:

```
S76543
        VERB  LOVE
        AGT    JOHN
        RECIP SALLY
```

to represent the assertion that John loves Sally. Here we have a situation of the same link names meaning different things depending on the nodes which they are connected to.

Without some explicit indication in the network notation that the two nodes are of different types, no mechanical procedure operating on such a network would be able to handle these links correctly in both cases. With an explicit indication of node type and an explicit definition that the meaning of an arc depends on the type of the node to which it is connected (and how), such a procedure could be defined, but a network notation of this sort would probably be confusing as an explanatory device for human consumption. This is functionally equivalent, however, to an alternative mechanism using a dual set of links with different names (such as R-AGT and AGT, for example) which would make the difference explicit to a human reader and would save the mechanical procedure from having to consult the type of the node to determine the import of the link. Notice that in either case we are required to make another extension of the semantics of our network notation since we have two different kinds of links with different kinds of import. The ones which make statements about possible slot fillers have assertional import (asserting facts about the predicate LOVE in this case) while the ones that make up the arguments of S76543 have structural import (building up the parts of the proposition, which incidentally may itself not be asserted but only part of some intensional representation).

We conclude that the difference between the specification of possible slot fillers for a predicate as part of the information about the predicate and the specification of particular slot fillers for particular instances of the predicate requires some basic distinction in our semantic network notation. One is left with several questions as to just how this distinction is best realized (for example does one want a dual set of link names -- or is there a preferable notation?) For the moment, however, let us leave those questions unexplored along with many issues that we have not begun to face and proceed with another problem of knowledge representation that imposes new demands on the interpretations of links and the conventions for representing facts in semantic networks.

6. Representing Quantified Expressions

The problem of representing quantified information in semantic networks is one that few people have faced and even fewer handled adequately. Let me begin by laying to rest a logically inadequate way of representing quantified expressions which unfortunately is the one most used in implemented semantic networks. It consists of simply tagging the quantifier onto the noun phrase it modifies just as if it were an adjective. In such a notation, the representation for "every integer is greater than some integer" would look something like:

S11113

VERB	GREATER
ARG1	D12345
ARG2	D67890

D12345

NOUN	INTEGER
MOD	EVERY

D67890

NOUN	INTEGER
MOD	SOME

Now there are two possible interpretations of this sentence depending on whether or not the second, existential quantifier is considered to be in the scope of the universal quantifier. In the normal interpretation, the second integer depends on the first and the sentence is true, while a pathological interpretation of the sentence is that there is some integer which every integer is greater than. (Lest you divert the issue with some claim that there is only one possible interpretation taking the quantifiers in the order in which they occur in the sentence consider a sentence such as "Everybody jumped in some old car that had the keys in it", in which the normal interpretation is the opposite.) Since our semantic network notation must provide a representation for whichever interpretation we decide was meant, there must be some way to distinguish the difference. If anything, the representation we have given seems to suggest the interpretation in which there is some integer

that every integer is greater than. If we take this as the interpretation of the above notation, then we need another representation for the other (and in this case correct) interpretation -- the one in which the second integer is a variable entity dependent on the first.

To complicate matters even further, consider the case of numerical quantifiers and a sentence such as "three lookouts saw two boats". There are three possible interpretations of the quantifiers in this case. In the one that seems to correspond to treating the quantifier as a modifier of the noun phrase, we would have one group of three lookouts that jointly participated in an activity of seeing one group of two boats. However, there is another interpretation in which each of three lookouts saw two boats (for an unknown total number of boats between 2 and 6 since we aren't told whether any of them saw the same boats as the others) and still another interpretation in which each of two boats was seen by three men. We must have a way in our network notation to unambiguously represent all three of these possible interpretations. Quillian's [1968] suggestion of using "criterialities" on the arcs to indicate quantification will fail for the same reasons unless some mechanism for indicating which arguments depend on which others is inserted.

Before proceeding to discuss logically adequate ways of

dealing with quantification, let me also lay to rest a borderline case. One might decide to represent the interpretation of the sentence in which each of three men saw two boats, for example, by creating three separate nodes for the men and asserting about each of them that he saw two boats. This could become logically adequate if the appropriate information were indicated that the three men were all different (it is not adequate to assume that internal nodes are different just because they are different nodes -- recall the morning star/evening star example) and if the three separate facts are tied together into a single fact somewhere (e.g., by a conjunction) since otherwise this wouldn't be an expression of a single fact (which could be denied, for example). This is, however, clearly not a reasonable representation for a sentence such as "250 million people live in the United States", and would be a logical impossibility for representing universally quantified expressions over sets whose cardinality was not known.

A variant of this is related to the transient process account. One might argue that it is not necessary to represent a sentence such as "Every boy has a dog" as a unit but one can simply add an assertion to each internal node representing a boy. To be correct, however, such an account would require a network to have perfect knowledge (i.e., an internal node for every boy that exists in the world), a

practical impossibility. We cannot assume that the entities in our network exhaust those that exist in the world. Hence we must represent this assertion in a way that will apply to future boys that we may learn about and not just to those we know about at this moment. To do this we must be able to store an intensional representation of the universally quantified proposition.

6.1. Quantifiers as Higher Operators

The traditional representation of quantifiers in the predicate calculus is that they are attached to the proposition which they govern in a string whose order determines the dependency of the individual variables on other variables. Thus the two interpretations of our first sentence are:

(∀X/integer) (∃Y/integer) (GREATER X Y)

and

(∃Y/integer) (∀X/integer) (GREATER X Y)

where I have chosen to explicitly indicate in the quantifier prefix the range of quantification of the variable (see Woods [1967] for a discussion of the advantages of doing this -- namely the uniform behavior for both universal and existential quantifiers). In the question-answering systems that I have constructed, including the LUNAR system, I have used a slightly expanded form of such quantifiers which

uniformly handles numerical quantifiers and definite determiners as well as the classical universal and existential quantifiers by treating the quantifiers as higher predicates which take as arguments a variable name, a specification of the range of quantification, a possible restriction on the range, and the proposition to be quantified (which includes a free occurrence of the variable of quantification and which may be already quantified by other quantifiers). In this notation, the above two interpretations would be represented as:

```
(FOR EVERY X / INTEGER : T ;
     (FOR SOME Y / INTEGER : T ; (GREATER X Y)))
     and
(FOR SOME Y / INTEGER : T ;
     (FOR EVERY X / INTEGER : T ; (GREATER X Y)))
```

where the component of the notation following the ":" in these expressions is a proposition which restricts the range of quantification (in this case the vacuously true proposition T) and the component following the ";" is the proposition being quantified. This type of higher operator representation of quantification can be represented in a network structure by creating a special type of node for the quantifier and some special links for its components. Thus, we could have something like:

S39732

TYPE	QUANT
QUANT-TYPE	EVERY
VARIABLE	X
CLASS	INTEGER
RESTRICTION	T
PROP	S39733

S39733

TYPE	QUANT
QUANT-TYPE	SOME
VARIABLE	Y
RESTRICTION	T
PROP	S39734

S39734

TYPE	PROPOSITION
VERB	GREATER
ARG1	X
ARG2	Y

This is essentially the technique used by Shapiro [1971b], who is one of the two people I know of to suggest a logically adequate treatment of quantifiers in his nets. (The other one is Martin Kay, whose proposal we will discuss shortly.) This technique has an unpleasant effect, however, in that it breaks up the chains of connections from node to node that one finds attractive in the more customary semantic network notations. That is, if we consider our sentence about lookouts and boats, we have gone successively from a simple-minded representation in which we might have a link labeled "see" which points from a node for "lookout" to one for "boats", to a case representation notation in which

the chain becomes an agent link from "lookout" to a special node which has a verb link to "see" and a patient link to "boats", and finally to a quantified representation in which the chain stretches from "lookout" via an inverse CLASS link to a quantifier node which has a PROP link to another quantifier node which has a CLASS link to "boats" and a PROP link to a proposition which has a VERB link to "see". Thus our successive changes in the network conventions designed to provide them with a logically adequate interpretation are carrying with them a cost in the directness of the associative paths. This may be an inevitable consequence of making the networks adequate for storing knowledge in general, and it may be that it is not too disruptive of the associative processing that one would like to apply to the memory representation. On the other hand it may lead to the conclusion that one cannot accomplish an appropriate associative linking of information as a direct consequence of the notation in which it is stored and that some separate indexing mechanism is required.

6.2. Other Possible Representations

There are two other possible candidates for representing quantified information, one of which to my knowledge has not been tried before in semantic networks. I will call them the "Skolem function method" and the "lamdda abstraction method," after well-known techniques in formal

logic.

a) Skolem Functions

The use of Skolem functions to represent quantified expressions is little known outside the field of mechanical theorem proving and certain branches of formal logic, but it is a pivotal technique in resolution theorem proving and is rather drastically different from the customary way of dealing with quantifiers in logic. The essence of the technique is to take a quantified expression containing no negative operators in the quantifier prefix (any such can be removed by means of the transformations exchanging "not every" for "some not" and "not some" for "every not") and replacing all instances of existentially quantified variables with a functional designator whose function is a unique function name chosen for that existential variable and whose arguments are the universally quantified variables in whose scopes the existential quantifier for that variable lies. After this the existential quantifiers are deleted and, since the only remaining variables are universally quantified, the universal quantifiers can be deleted and free variables treated as implicitly universally quantified. For example, the expression $(\forall x)(\exists y)(\forall z)(\exists w)\ P(x,y,z,w)$ becomes $P(x,f(x),z,g(x,z))$, where f and g are new function names created to replace the variables y and w.

Notice that the arguments of the functions f and g in the result preserve the information about the universally quantified variables on which they depend. This is all the information necessary to reconstruct the original expression and is intuitively exactly that information which we are interested in to characterize the difference between alternative interpretations of a sentence corresponding to different quantifier orderings -- i.e., does the choice of a given object depend on the choice of a universally quantified object or not. Thus the Skolem function serves as a device for recording the dependencies of an existentially quantified variable. An additional motivating factor for using Skolem functions to represent natural language quantification is that the quantification operation implicitly determines a real function of exactly this sort, and there are places in natural language dialogs where this implicit function appears to be referenced by anaphoric pronouns outside the scope of the original quantifier (e.g., in "Is there someone here from Virginia? If so, I have a prize for him", the "him" seems to refer to the value of such a function.) We can obtain a semantic network notation based on this Skolem function analogy by simply including with every existentially quantified object a link which points to all of the universally quantified objects on which this one depends. This is essentially the technique proposed by Kay [1973].

It must be pointed out that one difficulty with the Skolem function notation which accounts for its little use as a logical representation outside the theorem proving circles is that it is not possible to obtain the negation of a Skolem form expression by simply attaching a negation operator to the "top". Rather, negation involves a complex operation which changes all of the universal variables to existential ones and vice versa, and can hardly be accomplished short of converting the expression back to quantifier prefix form, rippling the negation through the quantifier prefix to the embedded predicate and then reconverting to Skolem form. This makes it difficult, for example, to store the denial of an existing proposition.) It seems likely that the same technique of explicitly linking the quantified object to those other objects on which it depends might also handle the case of numerically quantified expressions although I am not quite sure how it would all work out -- especially with negations.

b) Lambda Abstraction

We have already introduced Church's lambda notation as a convenient device for expressing a predicate defined by a combination or a modification of other predicates. In general, for any completely instantiated complex assemblage of predicates and propositions, one can make a predicate of it by replacing some of its specific arguments with variable

names and embedding it in a lambda expression with those variables indicated as arguments. For example, from a sentence "John told Mary to get something and hit Sam" we can construct a predicate (LAMBDA (X) John told Mary to get something and hit X) which is true of Sam if the original sentence is true and may be true of other individuals as well. This process is called lambda abstraction.

Now, one way to view a universally quantified sentence such as "all men are mortal" is simply as a statement of a relation between a set (all men) and a predicate (mortal) -- namely that the predicate is true of each member of the set (call this relation FORALL). By means of lambda abstraction we can create a predicate of exactly the type we need to view <u>every</u> instance of universal quantification as exactly this kind of assertion about a set and a predicate. For example, we can represent our assertion that every integer is greater than some integer as an assertion of the FORALL relation between the set of integers and the predicate

(LAMBDA (X) (X is greater than some integer))

and in a similar way we can define a relation FORSOME which holds between a set and a predicate if the predicate is true for some member of the set, thus giving us a representation:

```
(FORALL INTEGER (LAMBDA (X)

    (FORSOME INTEGER (LAMBDA (Y) (GREATER X Y)))))
```

which can be seen as almost a notational variant of the higher operator quantifier representation. Notice that the expression (LAMBDA (Y) (GREATER X Y)) is a predicate whose argument is Y and which has a free variable X. This means that the predicate itself is a variable entity which depends on X -- i.e., for each value of X we get a different predicate to be applied to the Y's.

The use of this technique in a semantic network notation would require a special type of node for a predicate defined by the lambda operator, but such a type of node is probably required anyway for independent reasons (since the operation of lambda abstraction is an intellectual operation which one can perform and since our semantic network should be able to store the results of such mental gymnastics). The structure of the above expression might look like:

S12233

TYPE	PROPOSITION
VERB	FORALL
CLASS	INTEGER
PRED	P12234

P12234

TYPE	PREDICATE
ARGUMENTS	(X)
BODY	S12235

S12235

TYPE	PROPOSITION
VERB	FORSOME

CLASS	INTEGER
PRED	P12236

P12236

TYPE	PREDICATE
ARGUMENTS	(Y)
BODY	S12237

S12237

TYPE	PROPOSITION
VERB	GREATER
ARG1	X
ARG2	Y

V. Conclusion

In the preceding sections, I hope that I have illustrated by example the kinds of explicit understanding of what one intends various network notations to mean that must be made in order to even begin to ask the questions whether a notation is an adequate one for representing knowledge in general (although for reasons of space I have been more brief in such explanations in this paper than I feel one should be in presenting a proposed complete semantic network notation). Moreover, I hope that I have made the point that when one does extract a clear understanding of the semantics of the notation, most of the existing semantic network notations are found wanting in some major respects -- notably the representation of propositions without committment to asserting their truth

and in representing various types of intensional descriptions of objects without commitment to their external existence, their external distinctness, or their completeness in covering all of the objects which are presumed to exist in the world. I have also pointed out the logical inadequacies of almost all current network notations for representing quantified information and some of the disadvantages of some logically adequate techniques.

I have not begun to address all of the problems that need to be addressed, and I have only begun to discuss the problems of relative clauses and quantificational information. I have not even mentioned other problems such as the representation of mass terms, adverbial modification, probabilistic information, degrees of certainty, time and tense, and a host of other difficult problems. All of these issues need to be addressed and solutions integrated into a consistent whole in order to produce a logically adequate semantic network formalism. No existing semantic network comes close to this goal.

I hope that by focusing on the logical inadequacies of many of the current (naive) assumptions about what semantic networks do and can do, I will have stimulated the search for better solutions and flagged some of the false assumptions about adequacies of techniques that might otherwise have gone unchallenged. As I said earlier, I

believe that work in the area of knowledge representation in general, and semantic networks in particular, is important to the further development of our understanding of human and artificial intelligence and that many essentially correct facts about human performance and useful techniques for artificial systems are emerging from this study. My hope for this paper is that it will stimulate this area of study to develop in a productive direction.

REFERENCES

Carbonell, J.R., & Collins, A.M. (1974)
"Natural semantics in artificial intelligence," Proceedings of Third International Joint Conference on Artificial Intelligence, 1973, pp. 344-351. Reprinted in the American Journal of Computational Linguistics, 1974, Vol. 1, Mfc. 3.

Davis, M. (1958)
Computability and Unsolvability, New York: McGraw-Hill.

Fillmore, C. (1968)
"The case for case," in Bach and Harms (eds.), Universals in Linguistic Theory, Chicago: Holt, Rinehart and Winston.

Kay, M. (1973)
"The MIND System," in R. Rustin (ed.) Natural Language Processing, New York: Algorithmics Press.

Lindsay, R.K. (1963)
"Inferential memory as the basis of machines which understand natural language," in E.A. Feigenbaum & J. Feldman (eds.), Computers and thought, New York: McGraw-Hill.

Quillian, M.R. (1968)
"Semantic memory," in M. Minsky (ed.), Semantic information processing, Cambridge: MIT Press.

Quillian, M.R. (1969)
"The Teachable Language Comprehender: A simulation program and theory of language," CACM, Vol. 12, No. 8, pp. 459-476.

Quine, W.V. (1961)
From a Logical Point of View, Second Edition, revised, New York: Harper and Row.

Raphael, B. (1964)
"A Computer Program which 'Understands'," AFIPS Conference Proceedings, Vol. 26, 1964 Fall Joint Computer Conference, pp. 577-589.

Rumelhart, D.E., Lindsay, P.H., & Norman, D.A. (1972)
"A process model for long-term memory," in E. Tulving & W. Donaldson (eds.), Organization of memory, New York: Academic Press.

Schank, R.C. (1975)
Conceptual Information Processing, Amsterdam: North-Holland.

Shapiro, S.C. (1971b)
"A net structure for semantic information storage, deduction, and retrieval," Proceedings of the Second International Joint Conference on Artificial Intelligence, pp. 512-523.

Simmons, W. (1973)
"Semantic networks: Their computation and use for understanding English sentences," in R.C. Schank & K.M. Colby (eds.), Computer models of thought and language, San Francisco: Freeman.

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

Woods, W.A. (1967)
"Semantics for a Question-Answering System," Ph.D. Thesis, Division of Engineering and Applied Physics, Harvard University. (Also in Report NSF-19, Harvard Computation Laboratory, September, 1967.) (Available from NTIS as PB-176-548.)

Woods, W.A. (1973a)
"Meaning and Machines," in A. Zampolli (ed.), Computational and Mathematical Linguistics, Proceedings of the International Conference on Computational Linguistics, Pisa, Italy, August 1973, Leo S. Olschki, Florence, Italy.

Woods, W.A. (1973b)
"Progress in Natural Language Understanding: An application to Lunar Geology," AFIPS Conference Proceedings, Vol. 42, 1973 National Computer Conference and Exposition, pp. 441-450.

Woods, W.A. (1975)
"Syntax, Semantics, and Speech," in R.D. Reddy (ed.) Speech Recognition: invited papers presented at the 1974 IEEE Symposium, New York: Academic Press (in press).

Woods, W.A., R.M. Kaplan, and B. Nash-Webber (1972)
"The Lunar Sciences Natural Language Information System: Final Report," BBN Report No. 2378, June 1972. (Available from NTIS as N72-28984.)