EXPANDING THE UTILITY OF SEMANTIC NETWORKS THROUGH PARTITIONING

Gary G. Hendrix
Artificial Intelligence Center
Stanford Research institute
Menlo Park, California 94025

Abstract

An augmentation of semantic networks is presented in which the various nodes and arcs are partitioned into "net spaces." These net spaces delimit the scopes of quantified variables, distinguish hypothetical and imaginary situations from reality, encode alternative worlds considered in planning, and focus attention at particular levels of detail.

I. Introduction

Semantic networks as used by Simmons f 1973], Shapiro [1971], Rumelhart and Norman 11972], and Schank 11973! have proved to be very serviceable structures for the encoding of knowledge, offering such recognized advantages as a convenient bi-directional linkage between semantically related data and an inherent facility for handling deep conceptual case systems. However, in their conventional form, networks are rather clumsy constructs for the representation of quantification. This clumsiness is largely a result of the difficulties encountered in specifying the scopes of variables. Maintaining models of several possible futures which may be reached from a current state by following different sequences of events has also been a formidable problem in networks because of the difficulties of distinguishing where the model of one "future world" ends and another begins. Similarly, it has been difficult to determine where the model of reality ends and models of imaginary or hypothetical situations begin. As a uniform mechanism for handling these and other problems of conventional nets, this paper presents the notion of net partitioning.

II Basic Network Notions

A semantic network consists of a collection of nodes and arcs where each node represents an object (a physical object, situation, event, set) and each arc represents a binary relation. For example, in the net of Figure 1, the node 'ANIMALS' (single quotes delimit nodes) represents the set ANIMALS, the set of all animals. Likewise, node 'DOGS' represents the set of all dogs. The arc labeled "s" from 'DOGS' to 'ANIMALS' indicates that DOGS is a subset of ANIMALS. The e arc from node 'FIDO* to node 'DOGS' indicates that FIDO is an element of set DOGS and hence that FI1>0 is some particular dog. Node 'H* encodes an owning situation, an element of the situation set <own>. In particular, 'H' represents the ownership of the dog FIDO by boy ELMO from time tl until t2. These components of situation H are associated with 'H' through arcs. For example, the arc labeled "ownee" from 'H' to 'FIDO' indicates that FIDO is the #@ownee of situation H, where the notation "#@ownee of H" means the value (#) of the ownee attribute (@) of H.

III. The Notion of Net Partitioning

The idea of net partitioning is to separate the various nodes and arcs into units called spaces so that every node and arc is assigned to exactly one space and all nodes and arcs lying in the same space are distinguishable from those of other spaces. Nodes and arcs of different spaces may be linked, but the linkage must pass through "boundaries" which separate one space from another. It is possible that a node (arc) may lie in a different space from any of its associated arcs (nodes).

Ordinarily, spaces are used for such purposes as delimiting the scopes of quantified variables- However, before taking up such practical applications, consider the simpler (if atypical) partitioning shown in Figure 2. Each space is enclosed within a dotted line. For example, space SA is at the top of the figure and includes nodes 'ANIMALS', 'DOGS', '<live-on>' and 'FLEAS'. SA also includes the two s arcs ("subset") which indicate that set DOGS and set FLEAS are subsets of ANIMALS. Diagrammatically, an arc belongs to a space if the arc's label is written within the dotted line boundaries of the space. Thus the e arc ("element") from 'FERDINAND' to 'FLEAS' lies in S3.

The various spaces of a partitioning are organized into a lattice such as in Figure 3. Viewing the semantic net from some point S in this lattice, only those nodes and arcs are visible which lie in S or in a space above S in the lattice. For example, from space S2, only those nodes and arcs lying in S2 or SA are visible. In particular, it is possible to see that FIDO is a DOG and that DOGS are ANIMALS, but it is not possible to see that FERDINAND is a FLEA. From S5, information in S5, S3, S2, and SA is visible. Hence, from S5, the whole of the net of Figure 1 may be seen. (A space lattice always has a special top space SA and a special bottom space SZ. Information encoded in SA is visible from all spaces while the total net is visible from SZ.) Partitioned networks are constructed by creating empty net spaces, adding them to the lattice, and then creating nodes and arcs within each newly created space.

iv.Using Partition Spaces to Encode Quantification

A primary reason for imposing partitioning upon semantic nets is to delimit the scopes of quantified variables. In considering quantified statements, it will be helpful to build upon the network encoding of specific pieces of information.

Recording a Specific Event

As an example of the network encoding of a single event, consider

"A dog 1) bit a postman P,"

encoded by the net of Figure 4. Nodes 'DOGS', '<bite>' and 'POSTMEN' represent the sets of all dogs, biting events and postmen respectively. Node 'D' with e arc to 'DOGS' represents a particular dog D. Similarly, 'B' represents a particular biting event and 'P ' represents a particular postman. B, like other biting events, has an internal structure consisting of an #@assailant (the value (#) of the bite's assailant attributed)) and a #@victim. The #@assailant and #©victim of B are D and P respectively. (Note: time information has been deleted in this and other examples for the sake of simplicity, but may be indicated in the network by including time arcs. A network treatment of time is discussed in Hendrix [forthcoming].)

Existential Quantification

The example "a dog D bit a postman P" may be paraphrased as "There exists a dog D and a postman P such that *D* bit P" which resembles the predicate calculus statement

$$(\exists D \ \epsilon \ \text{DOGS}) \ (\exists P \ \epsilon \ \text{POSTMEN}) \ [\text{BIT}(D, P)]$$

In predicate calculus, dog D and postman P are represented as objects while the biting event is expressed by a predicate. But the biting event is a situation which might be the object of some more complex event such as a "wanting (as in Section VI below). Considering the biting event to be just as much an object as the dog and postman, the example sentence may be paraphrased as "There exists a dog D and a postman P and a biting event B. Further, the following relationships exist: <D is the #@assailant of B>, and <P is the #@ victim of B>."

Following this paraphrase, the presence of each node and arc within space SA may be interpreted as an implicit statement of existence. Thus, the nodes in Figure 4 assert that there exist objects DOGS, <bite>, POSTMEN, D, B and P. The arcs assert that there exist relationships <D ε DOGS>, <B ε <bitev>, <p ε POS1MEN*>, <D is the #©assailant of B>, and <P is the #@victim of B>.

Universal Quantification

While "a dog bit a postman" asserts the occurrence of only one event, a universally quantified statement such as

"Every dog has bitten a postman"

asserts the existence of many separate events. Further, the statement indicates how the several events follow one recurring pattern. An ability to encode universally quantified statements is of considerable importance since it is often impractical (or impossible) to record the same information by a collection of individual explicit statements because of the very number (possibly infinite) of statements required.

Using partitioning, the statement above is encoded by the net of Figure 5. Node 'GS' (a node especially known to utility routines) represents the set of all general statements (the set of statements involving universal quantifiers or, under another interpretation, the set of recurring patterns of events.) Node 'g * represents the particular statement (set of events) cited above. Thus, g is the fact that "every dog has bitten a postman."

Characteristically, a general statement encodes a collection of separate situations all of which follow the same basic pattern. This pattern is represented by the #@form of the general statement. The #@form of g is encoded by space Si, which lies just below SA in the lattice. S1 is regarded as a super-node containing its own internal structure and representing a composite variable that takes on a different value for each instantiation of the recurring pattern. Each node and arc within S1 is regarded as a subvariable.

General statements are also typically associated with universally quantified variables that assume values from some specified range. Statement g has a universally quantified variable d given by its @*Vv* attribute. Note that variable d is necessarily a part of the #@aform of g (i.e., *d' lies in space S1). From *d* there is an e arc to 'DOGS* indicating that the value of d (written #d) must be taken from the range set DOGS.

The interpretation of a general statement is that for each assignment of the variables #@Wv to values in their corresponding ranges, there exist entities matching the structure of the #@form. For g this means that for every #d ε DOGS there exist #bEF<bite>, #p ε POSTMEN and the relations <#d is the #@assailant of #b> and <#p is the #@victim of #b->. In particular, if d assumes the value D, and p (a function of d) assumes the value P, then the #@form of g assumes as its value roughly the net of Figure 4.

It is useful to think of the space which encodes a #$>form as representing a hypothetical world which is mapped into reality for each assignment of values to the #@*Vv* variables. That is, for each assignment of the *Wv variables, there exists a "copy" of the #@form in reality.

The space encoding the #$>form serves to delimit the scope of the statement's universally quantified variables. Although other methods for indicating scopes within nets have been used (e.g., Shapiro tI97lI), spaces are particularly well suited to this purpose in that the node and arc variables encoding information within a #@form look exactly like the representations of specific facts. More importantly, such variables are effectively isolated from "real" information by partition boundaries (Section V). Without partitions, variables may become confused with the values they represent unless cumbersome markers (in the form of modified arc and node labels, special flags or extra structural arcs) are added to the net as preventatives.

Variations *of* the Example Statement

The network of Figure 6 encodes "Every dog in town has bitten the constable" (or "There exists a constable who has been bitten by every dog in town"). In Figure 5, the bitten postman is a (Skolem) function of the biting dog. Hence, the postman is encoded by a node 'p' lying in the #!@form of g. This puts p (and, in general, any variable lying in the #$>form) within the scope of the universal quantifier. In contrast, the constable C of Figure 6 is encoded by node 'C' lying outside the #@form of g*. Hence, all the biting events involve the same constable C. The victim arc remains within the scope since each biting is associated with a different instance of the victim relation between a (new) biting and the (one) constable.

As a second variation, consider "Every dog has bitten every postman," encoded by the net of Figure 7. This statement involves two universally quantified variables d and p. For each combination of value assignments to d and p, there exists a biting event ffb whose #@assailant is the value of d (#d) and whose #@WvIctim is #p.

Alternating Quantification

Complex quantifications involving nested scopes may also be encoded by nesting spaces, as shown abstractly in Figure 8. Space SI encodes the scope of universal variable a while S4 encodes the scope of the universal variable c.

V. The Invisibility Feature

By lying in a separate space, nodes and arcs of a general statement's #@form are invisible unless the statement is being deliberately examined. The importance of this invisibility feature is illustrated by Figure 9. If a need arises to locate all explicit representations of DOGS, an algorithm can start at 'DOGS' and look for incoming e or s arcs. Such an algorithm leads eventually to the discovery of 'Fido' and 'Muttley'. It must not be allowed to discover 'd', however, since '*d ' is a variable representing no dog in particular.

But the variable 'd ' and its e arc lie in a separate space S1. If the search for explicit representations of dogs is conducted in SA or one of its descendants in the lattice which is not also a descendant of S1, then the e arc from node 'd ' will never be seen, and variable 'd' will not be erroneously considered as the representation of a particular dog. Net partitioning thus allows information concerning general statements and particular situations to be mixed freely and represented homogeneously while avoiding any confusion of one type of information with the other.

While information contained within the #@form of a general statement is invisible under "normal" conditions, it may be made visible whenever needed by viewing the network from SZ, the bottom space in the space lattice. Viewing the net from SZ allows general statements relevant to a given query to be retrieved and (hopefully) used to deduce an appropriate response.

VI. Other Uses of Partitioning

The partitioning that distinguishes variables from explicit representations may also be used to set apart portions of the net for a variety of other purposes.

One such purpose is to distinguish hypothetical situations from their real world counterparts and from each other. For example, consider how the statement

"Fido wanted to bite a (any) postman"

is encoded by the network of Figure 10. Node 'W represents a wanting situation in which Fido is the #@wanter and the thing wanted (the #@wanted) is represented by net space SW. Space SW encodes information about a hypothetical world that the #@wanter desires to be mapped into real existence. Specifically, Fido wants the existence of a biting situation #b, a postman #p and the relationships <-Fido is the #©assailant of #b> and <#p is the victim of #b>. Although there exist links (such as the e arcs and the assailant arc) from the hypothetical world of SW into reality, the conditions in SW are effectively isolated from the model of reality by the partition boundary.

The ability to quarantine "unreal" information within hypothetical spaces has many other applications. For example, Fido's dreams, fantasies, beliefs and even the lies he tells may be encoded within such spaces. Without net partitioning, it is difficult to determine which nodes and arcs encode hypothetical information and which reality. Harder still is the distinction of one hypothetical world from another.

For planning applications, hypothetical spaces may be used to encode the alternative world models set up to depict states of the world which may be achieved by performing various sequences of operations from an initial state. A planner using partitioned nets could share a common data base with a network language processor.

Parsing, especially the parsing of continuous speech, has much in common with planning in that any number of alternative hypotheses may be under consideration at any one time. Alternative interpretations of utterance fragments may be maintained in separate spaces. If alternative phrases PI and P2, represented respectively by spaces SI and S2, share a common subphrase PO, represented by SO, then SI and S2 will lie below SO in the partition lattice. Thus, the structures constructed to represent PO will be visible from both SI and S2 and need be built only once.

VII. All Existing System

Partitioned semantic networks of the kind described above have been implemented for the speech understanding system being developed jointly by SRI and *SDC* (see Walker [1975 J). As an application of the quantified statements discussed above, the speech system uses spaces to encode rules defining categories of objects. The taxonomic system based on these categories compresses redundant information by recording common data at the category level rather than with each individual. Those network category rules which define the attribute values associated with various verb classes aid the translation routines by providing information needed to determine if a parse is semantically well formed and to associate surface structures with their corresponding network representations.

The speech understanding system also uses spaces to maintain alternative parsings and in focusing the system's attention on a particular portion (context) of the semantic network for discourse analysis.

VIII. Relationship to Other Partitioning Systems

A number of modern computer languages, exemplified by QLISP (Reboh [1973]), have data base maintenance facilities with built-in mechanisms for partitioning data into "contexts." However, all such systems which are known to the author are based on assertional representations as opposed to networks. The context hierarchies of these systems are strictly tree-like and are not allowed to form lattices. Further, the context facility of these systems has neither been exploited as a tool for implementing quantification nor been used in event descriptions for encoding the objects of such actions as wanting and dreaming.

Acknowledgements

The author is indebted to R. F. Simmons for his counsel during the early stages of this research and to the members of the SRI speech understanding and QLISP groups for their aid in solidifying and implementing the concepts herein presented. The research reported in this paper was supported at SRI by the Advance Research Projects Agency under Contracts DAHC04-72-C-0008, DAHC04-72-C-0009, DAHC04-75-C-0004 and DAHC04-75-C-0005 with the U.S. Army Research Office.

References

Hendrix, G.G., "Partitioned Networks for the Mathematical Modeling of Natural Language Semantics," dissertation In progress, Computer Sciences Department, Univermity of Texas, Austin.

Rumelhart, D.E. and D.A. Norman, Active Semantic Networks as a Model of Human Memory, Advance Papers, Third International Joint Conference on Artificial Intelligence, Stanford, California, (1973) 450-457.

Schank, R.C., Identification of Conceptualizations Underlying Natural Language. In Schank and Colby (eds.), Computer Models of Thought and Language, W.H. Freeman and Co., San Francisco, California (1973).

Shapiro, S.C., A Net Structure for Semantic Information Storage, Deduction and Retrieval, Advance Papers, Second International Joint Conference on Artificial Intelligence, London, (1971) 512-523.

Simmons, R.F., Semantic Networks: Their Computations and Use for Understanding English Sentences. In Schank and Colby (eds.), Computer Models of Thought and Language, W.H. Freeman and Col, San Francisco, California (1973).

Walker, D.E., et al., Speech Understanding Research, Annual Report, Project 3804, Artificial Intelligence Center, SRI, Menlo Park, Ca (1975).

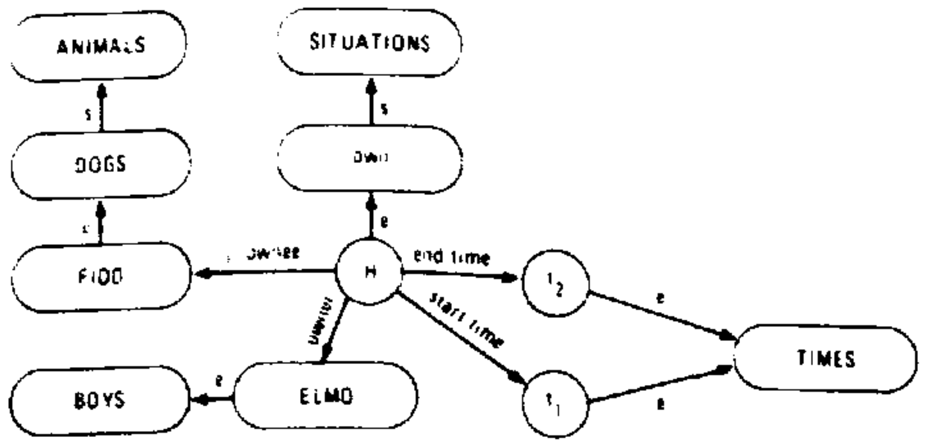

FIGURE 1 A TYPICAL NET FRAGMENT

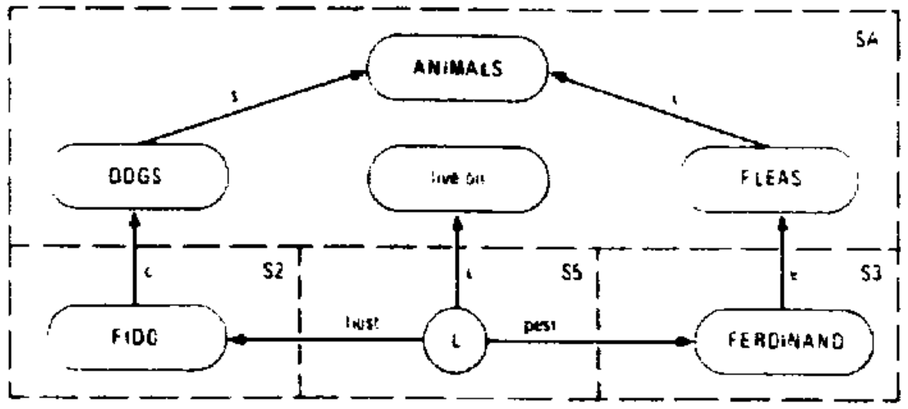

FIGURE 2 A SAMPLE NET SPACE PARTITION

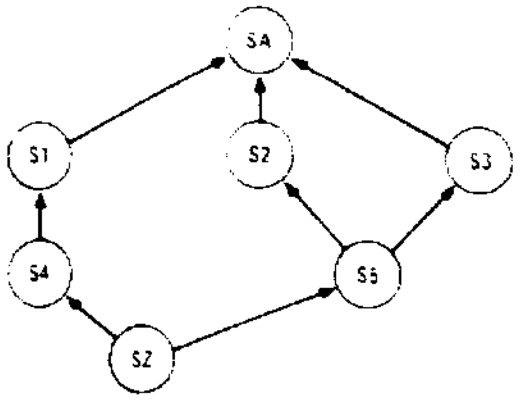

FIGURE 3 A SAMPLE NET SPACE LATTICE

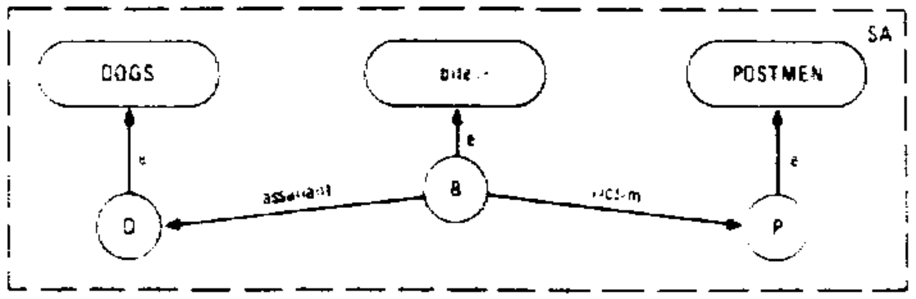

FIGURE 4 THE DOG BIT THE POSTMAN

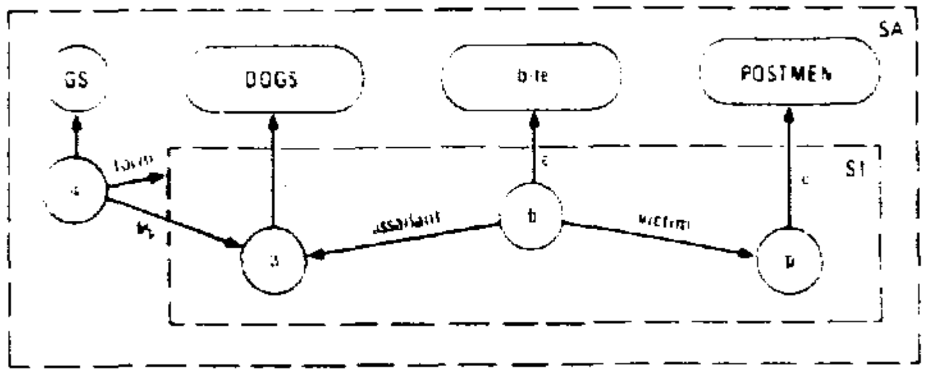

FIGURE 5 EVERY DOG HAS BITTEN A POSTMAN

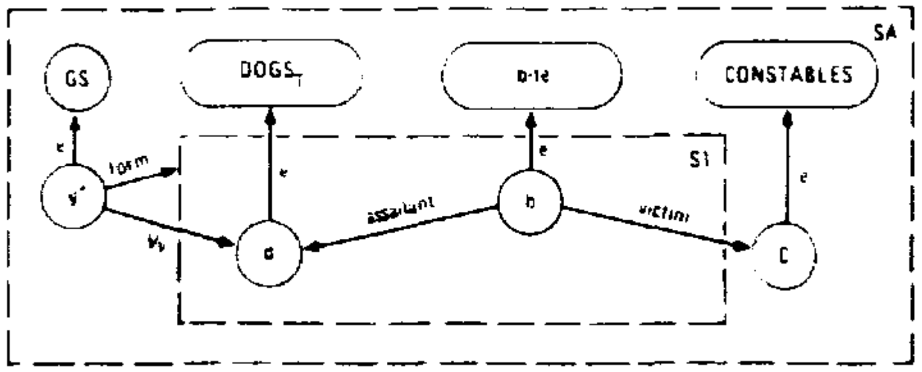

FIGURE 6 EVERY DOG IN TOWN HAS BITTEN THE CONSTABLE

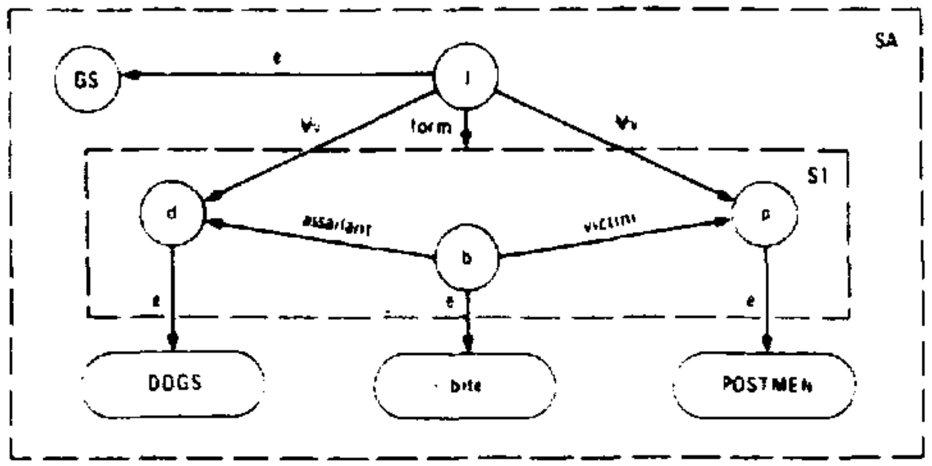

FIGURE 7 EVERY DOG HAS BITTEN EVERY POSTMAN

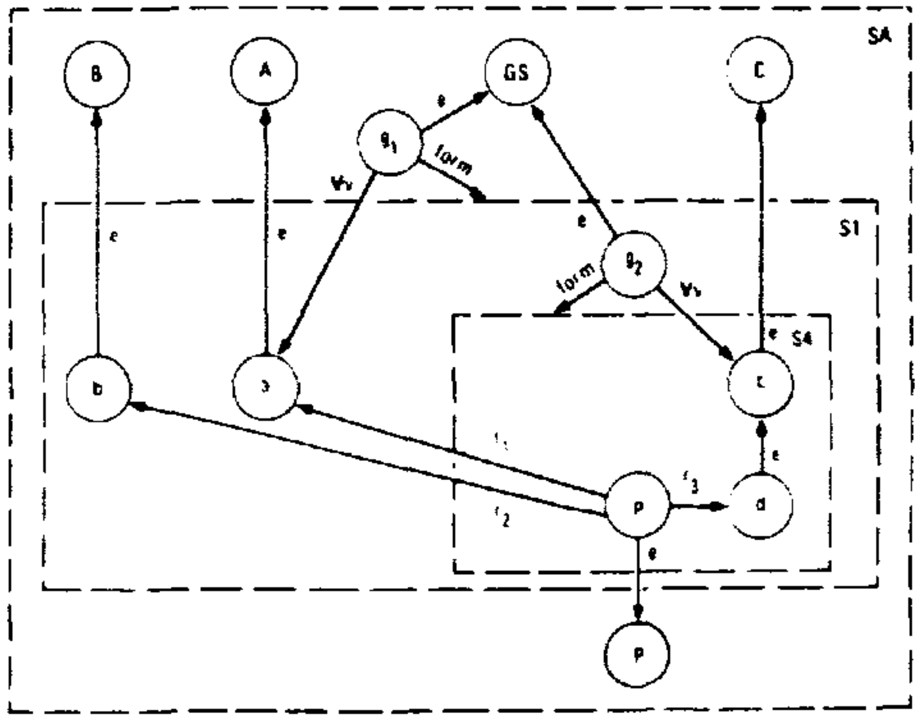

FIGURE 8 $(\forall a \in A)(\exists b \in B)(\forall c \in C)(\exists d \in c)\ P(a, b, d)$

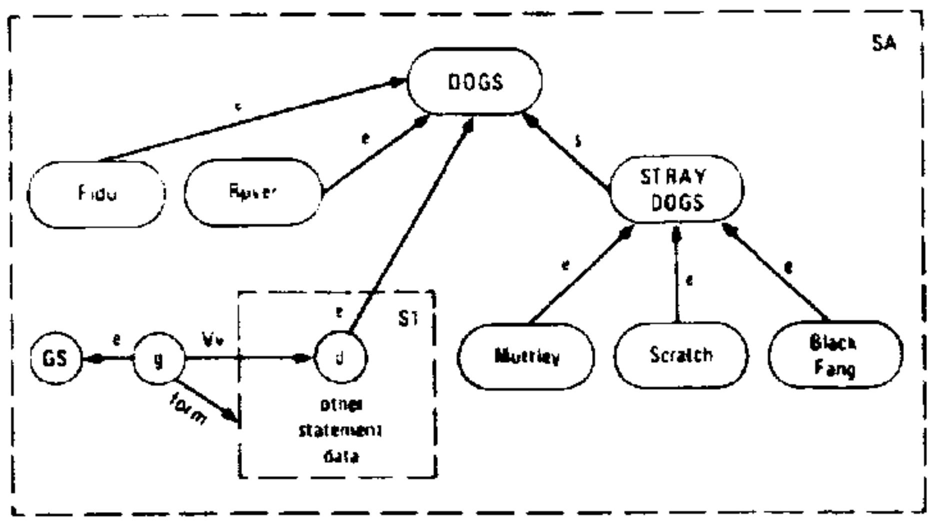

FIGURE 9 VISIBLE AND INVISIBLE NODES

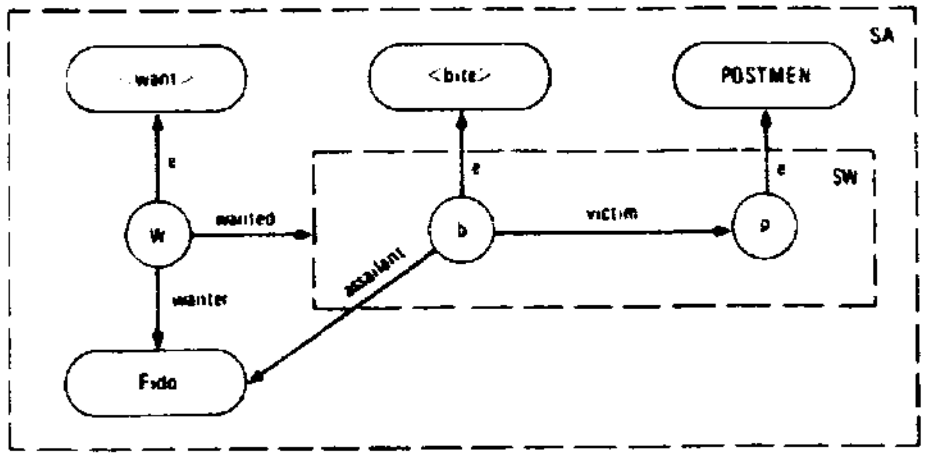

FIGURE 10 FIDO WANTED TO BITE A POSTMAN