ENCODING KNOWLEDGE IN PARTITIONED NETWORKS

Technical Note 164

June 28, 1978

By: Gary G. Hendrix
Artificial Intelligence Center

SRI Project 5844

To be published in Associative Networks-The Representation and Use of Knowledge in Computers, Nicholas V. Findler (ED.), Academic Press,New York.

The work reported herein was supported by the National Science Foundation under Grant No. MCS76-22004, the Defense Advanced Research Projects Agency under Contracts DAHC04-75-C-0005, DAAG29-76-C-0011, DAAG29-76-C-0012 and N00039-78-C-0060, and the Office of Resources of the U. S. Geological Survey under Contract No. 14-08-0001-15985.

ABSTRACT

This paper discusses network notations for encoding a number of different kinds of knowledge, including taxonomic information; general statements involving quantification; information about processes and procedures; the delineation of local contexts, beliefs, and wishes; and the relationships between syntactic units and their interpretations.

Many of the encodings appeal to the concept of network partitioning, in which a large net is partitioned into subnets and higher-order relationships among the subnets are defined.

Procedural mechanisms for constructing and using the various network formalisms are discussed as equal partners with the declarative structures.

I INTRODUCTION

Over the past three years, several systems* have been constructed in SRI International's Artificial Intelligence Center that make use of partitioned network** structures as a medium for recording knowledge. These systems perform such diverse tasks as translating natural language into formal structures, performing logical deduction, doing judgmental reasoning, reasoning about the structure of data in data bases, reasoning about processes, interrelating the sentences of a dialogue, and generating natural language descriptions of information that is stored in formal structures. This paper looks at the network techniques used in the various projects from a uniform perspective, describing both the encoding techniques that are common to most of the systems and the special techniques devised to handle specialized tasks.

----

* A system for judgmental reasoning is described in [Duda, Hart, Nilsson and Sutherland 1977] and in [Duda, Hart and Reboh 1977]. Systems for deduction, discourse analysis, natural language understanding and natural language generation are discussed in considerable detail in [Walker 1978]. By permission from its publisher, some of the examples and figures of the latter work are reproduced herein.

** Although I have used the term "semantic network" in the past, it is my intention to avoid its use henceforth. The term "semantics" is best used to refer to the relationship between linguistic structures (words, phrases, sentences, discourses) and their meanings. Because the networks described here are used primarily (but not exclusively!) to encode the knowledge conveyed by language, rather than the relationship of language to what it conveys, the term "knowledge network" or "K-net" seems more appropriate.

II BACKGROUND AND MOTIVATION

A. Why Use Nets

Before plunging into the details of how networks can encode information, it is worthwhile to reflect on the general reasons for selecting nets as a representation medium. Their attraction largely centers around two factors. First, it is believed that the expressive power of nets is sufficient to encode any fact or concept that is encodable in any other formal, symbolic system. This means that nets may serve as a common medium of representation for diverse kinds of knowledge. Second, and this is the point that distinguishes network structures from other formally complete systems, the network data structures that encode information may themselves serve as a guide for information retrieval. From a given node, nodes representing related entities are found simply by following pointers from the node to its neighbors. In this way, a network provides its own meaning-bearing indexing system. To the extent that the labels on arcs and nodes are meaningful to net-manipulating procedures, they provide guidance to help traverse the net in search of information relevant to a task.

B. Partners with Nets

The knowledge encoded in a network, being declarative, is somewhat like that stored in a book: it is available for the support of intellectual activity only if there exists some outside agent that can retrieve the knowledge and apply it. This outside agent embodies knowledge about how to manipulate the information in the net, and may have access to yet other bodies of information. To the extent that the information in the network is to be used, the network and its manipulator are mutually dependent partners. Therefore, in considering the network structures presented below, it is important to consider also the procedures that manipulate them.

C. Networks as a Medium for Integrating Skills

Just as the knowledge in a book may be accessed and applied by multiple agents, so may the knowledge in a network. In particular, information in a net may be used by a number of different procedures in performing a variety of tasks. For example, constraints on set membership recorded in a network may aid both the process of natural language understanding and the process of logical deduction.

But a net need not merely provide a static repository of information that is to be shared by multiple processes. Rather, it may serve as a medium of communication between processes. For example, natural language translation may create a network description of a question. A question answerer may then process the question's description to produce a network-encoded answer, which in turn is the input to an English generator. All of the intermediate structures may be examined by discourse analysis procedures that seek to build up a network-encoded model of an extended dialogue.

The point is that their representational power makes nets attractive both as a medium for encoding information needed by multiple skill modules and as a common language for communicating results among modules. Thus, nets may aid in two ways in the integration of multiple intelligence skills.

III BASIC NETWORK NOTIONS

This section introduces the basic techniques used by our systems to encode information in networks. The representation builds upon such work as that reported in [Simmons 1973], [Shapiro 1971] and [Norman, Rumelhart and the LNR Group 1975], but is tied more closely to the notions of conventional logic than are most network systems.*

----

* The networks of Schubert [Schubert 1976] and Kay [Kay 1973] are also closely tied to conventional logic, but do not use partitioning.

A. A Preliminary Example

In its simplest form, a knowledge representation network consists of a collection of nodes interconnected by an accompanying set of arcs. Each node denotes an object (a physical object, situation*, set) and each arc represents an instance of a binary relation. For example, the nodes JOHN and MEN in Figure 1 denote a man John and the set of all men, respectively. The arc labeled "e" from JOHN to MEN indicates that John is an element of the set of men and is thus some particular man.

Further details concerning how the interconnections among nodes and arcs can be used to encode knowledge may be seen by considering Figure 1 systematically. At the top of the figure is the node UNIVERSAL. This node denotes the set Universal, the universal set of objects.** Arcs labeled "s", called "s arcs", are used to indicate subset relationships that exist between Universal and other sets. In particular, the s arc from HUMANS to UNIVERSAL indicates that Humans, the set of all human beings, is a subset of Universal. Similarly, Situations, Times, and Physobjs (the set of all physical objects) are also indicated as being subsets of Universal. At the next lower level, Men is shown to be a subset of Humans.

As indicated above, set membership is encoded in the network through the use of "e arcs". Thus, the network of Figure 1 indicates that John is a man, Old.Black is an automobile, and T1 and T2 are instants in time.

The node Q denotes an element of the set Ownings, the set of

----

* McCarthy [McCarthy 1968] has used the term <u>situation</u> to refer to the complete state of affairs at some instant of time. I prefer to use the term <u>state of the world</u> to refer to this concept, and to use the term <u>situation</u> to refer to any event or state of being that occurs over an interval of time. Thus, my notion of a situation might be applied to such conditions or circumstances as Mary owning a car (a state of being), or Mary driving her car to town (an event).

** Symbols composed of all capital letters are used as the names of nodes. Entities denoted by nodes are given names in which only the first letter is capitalized.

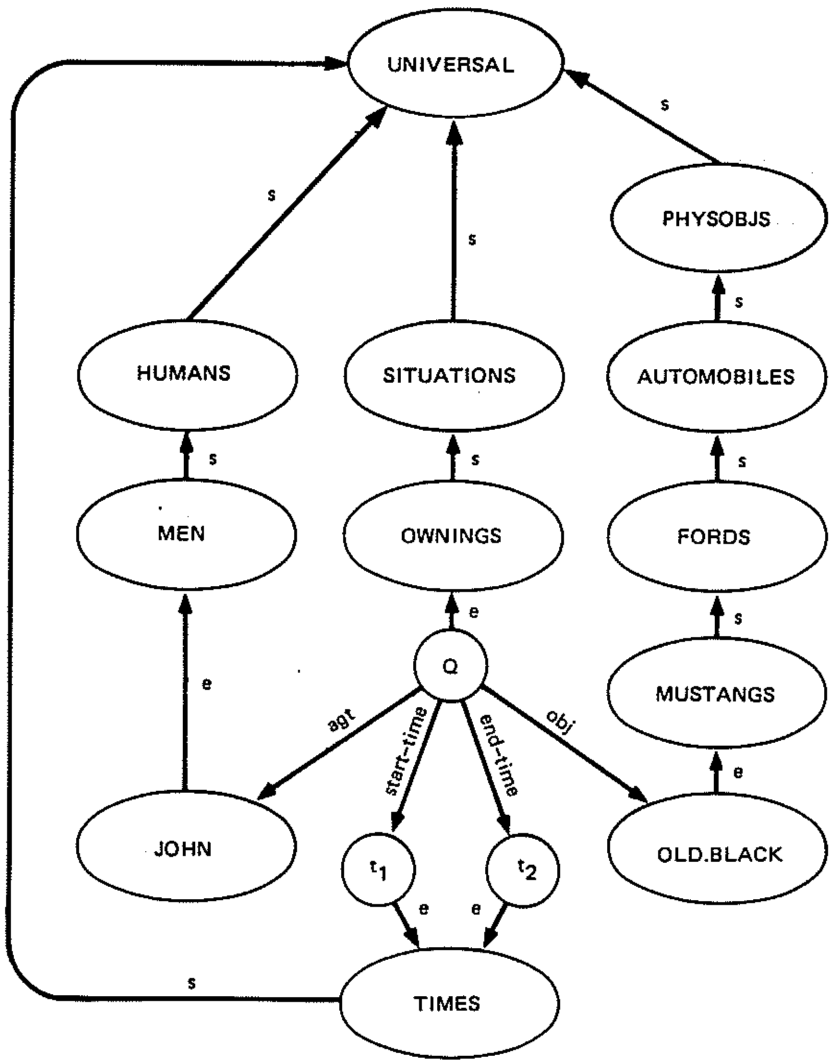


FIGURE 1 A SIMPLE REPRESENTATION NETWORK

situations in which an agent owns an object over some time period.[+] In turn, Ownings is a subset of Situations, which is the set of all static conditions and dynamic events. For the particular situation Q, John is the agent that owns the object, Old.Black, during the period from time T1 until T2. The components of situation Q are associated with it through deep case relationships. In general, a deep case (or slot or role) is a relationship between a situation (or other composite object) and a participant in the situation. For example, the agent of situation Q is indicated by the agt arc from Q to JOHN. (The notion of a deep case, which is a relationship between world objects, contrasts in the linguistic literature with the notion of a surface case, which is a relationship between syntactic units.*)

B. Useful Restrictions on Nodes and Arcs

Proponents of network structures have adopted a number of different conventions concerning what types of concepts may be encoded by nodes and what types of relationships may or should be encoded by arcs.** In creating our encoding structures, we have attempted to use constructs that are understood by appealing to such familiar mathematical systems as set theory and predicate calculus. Our nets place no restrictions on the types of objects that may be represented by nodes. However, arcs are restricted to the encoding of formal binary relationships, such as taxonomic (element and subset) relationships and deep case relationships.*** Deep case relations must be functions. So, for example, any Ownings situation has exactly one start-time, but the same time T may be the start-time of many situations. Deep cases must also

----

+ Methods for describing properties that are common to all elements of a set are discussed subsequently.

* See [Fillmore 1968].

** For a useful perspective on these issues, see [Woods 1975].

*** The set of case relationships is open-ended. In particular, no fixed set such as {AGT, OBJ, GOAL, THEME} is assumed.

be constant over time and circumstance. Arcs are never, for example, allowed to encode relationships, such as ownership, that are time bounded.

Relationships that are not represented by arcs are represented by nodes having outgoing case arcs pointing to the participants in the relationship (such as node Q in Figure 1). This representational convention allows an arbitrary amount of information to be stored with a relationship (using case arcs) and allows associative retrieval of the relationship using the network's indexing facilities (i.e., by following the arcs). Such relationships are grouped by type into sets and these sets are considered to be subsets of the set of all situations.

Some network systems have a small fixed number of arc labels, with each having a special meaning to the network processor. However, having many meaningful labels can be quite beneficial. Even our general-purpose routines that retrieve information from a network without special knowledge concerning the meanings of arc labels operate more efficiently when more case names are used, because an increase in the number of arc labels provides a finer index into the net.

C. The Hierarchical Taxonomy

The presence of e and s arcs in a network serves to taxonomize the concepts represented by the various nodes in hierarchical form and is a key feature of the notation.* Because the knowledge of whether or not an item belongs to a given set is of central relevance in question answering and fact retrieval, the taxonomy itself often provides a natural and concise expression of major portions of the information about a task domain. The significance of the taxonomy is further enhanced by the fact that the members of many sets have a collection of properties in common. Any property that is characteristic of <u>all</u> members of a given set may be described at the set level and need not be repeated in the encoding of each individual set member. This set-level

----

* By using disjunction, certain ambiguities regarding the hierarchy may be encoded. For example, it is easy to represent the fact that John's pet is either a dog or a cat.

encoding, which requires the use of universal quantification, leads to great savings in storage.

To enhance the precision of the network encoding of taxonomies, the standard set-theory notions of set membership and set inclusion, which are expressed by e and s arcs, may be supplemented by the more restrictive concepts of disjoint subsets and distinct elements.

Most sibling subsets described in taxonomies are disjoint. Arcs labeled "ds" are used to represent this disjointness property in a concise and easily interpretable manner. A ds arc from a node X to a node Z indicates that the set denoted by X is a subset of the set denoted by Z and that the X set is disjoint from any other set denoted by a node with an outgoing ds arc to Z. For example, the ds arcs in Figure 2 emanating from the HUMANS and COMPANIES nodes indicate that the set of Humans and the set of Companies are disjoint subsets of the set of Legal-Persons.

Since each node in most taxonomies denotes a distinct entity, and in general an entity can be denoted by any number of nodes, arcs labeled de (for "distinct element") are used to indicate that each of two or more nodes denotes a different element of a set. In particular, a de arc from a node X to a node Z indicates that the entity denoted by X is an element of the set denoted by Z and that the X entity is distinct from any other entity denoted by a node that has an outgoing de arc to Z. For example, the de arcs in Figure 2 emanating from G.M. and FORD indicate that G.M. and Ford are distinct members of the set of companies.

To see the useful interplay between de arcs and e arcs, suppose Tom, Dick, and Harry went for a drive, and the driver wore a red cap. Tom, Dick, and Harry are distinct elements of the set of people who went for the drive, and their membership in the set would be recorded by three de arcs. The driver is also in this set, but could be any one of the three. Using a normal e arc to show the membership of the driver allows information about the driver (e.g., he wore a red cap) to be recorded while maintaining the uncertainty as to which of the three set members the driver really is.

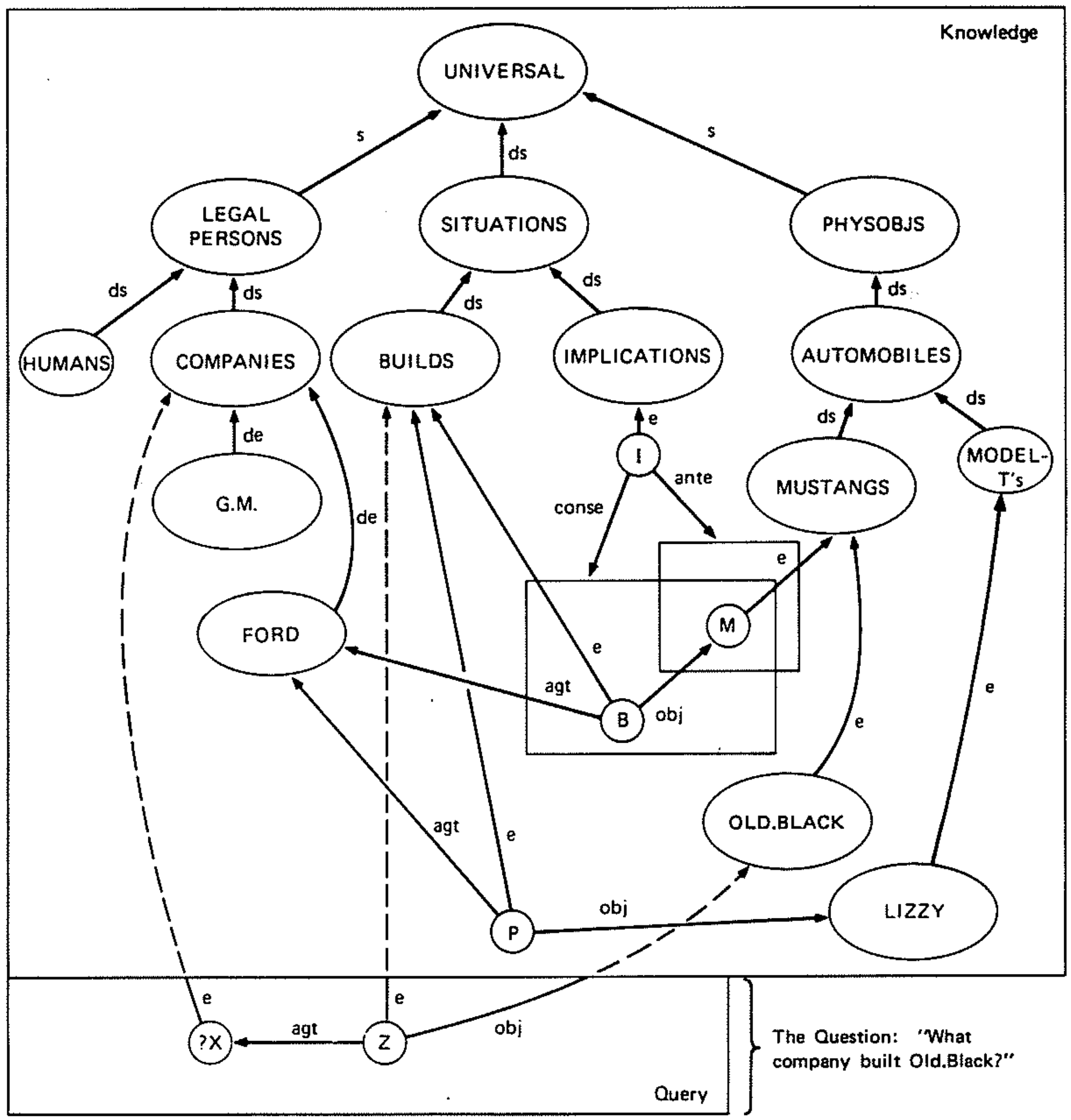


FIGURE 2 WHAT COMPANY BUILT OLD.BLACK?

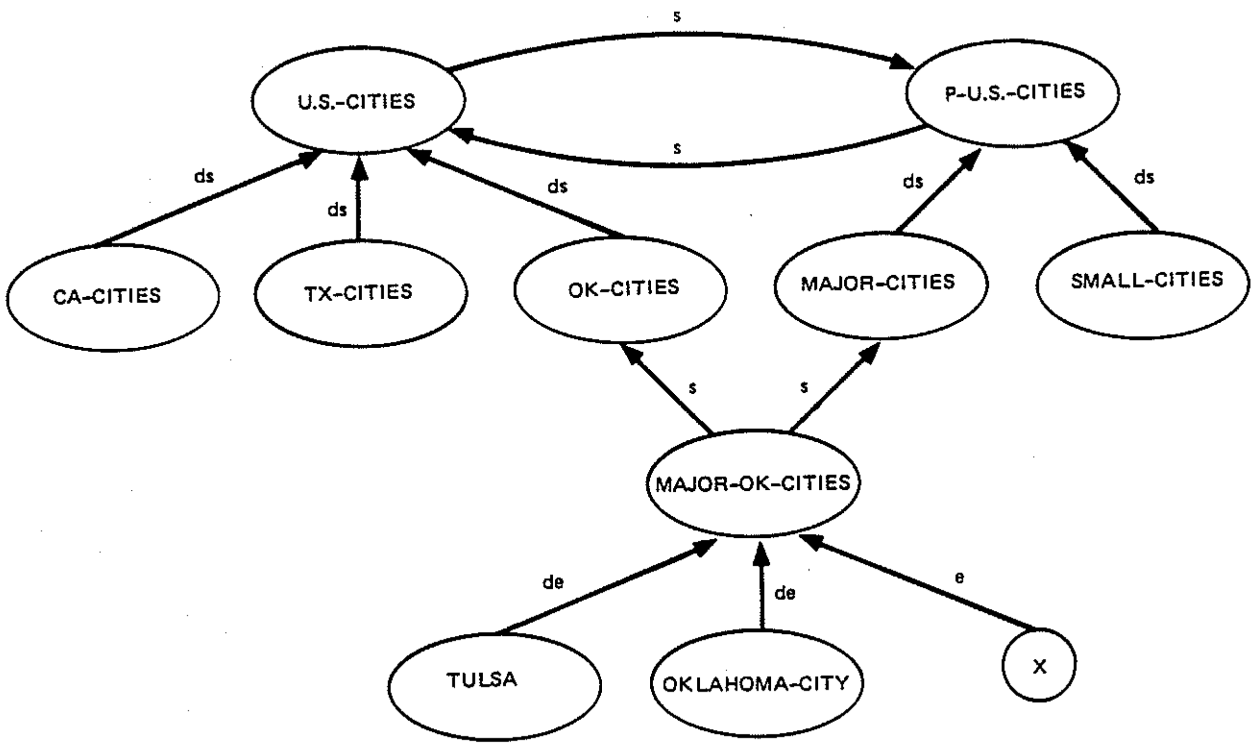


Figure 3 Taxonomy of U. S. Cities

The use of e, s, de, and ds arcs in a more extended example is shown in Figure 3. The network indicates that US-Cities and P-US-Cities are both subsets of each other. Hence, the nodes US-CITIES and P-US-CITIES may both be interpreted as denoting the set of all cities in the USA. The node US-CITIES is used to help taxonomize cities by state. The ds arcs to US-CITIES from CA-CITIES, TX-CITIES, and OK-CITIES indicate that the sets of cities in California, Texas, and Oklahoma are all subsets of US-Cities and are disjoint from one another. The node P-US-CITIES is used to help taxonomize cities by population into the disjoint sets Major-Cities and Small-Cities. Notice particularly that any of the disjoint subsets of P-US-Cities <u>may</u> (but in general need not) intersect with any of the disjoint subsets of US-Cities. In particular, the network shows Major-OK-Cities to be a nonempty subset of both OK-Cities and Major-Cities.

The membership of Major-OK-Cities includes Tulsa, Oklahoma-City, and X. Tulsa and Oklahoma-City are shown to be distinct. X might be either of these or yet some other city. If the cardinality of Major-OK-Cities is 2, then it is possible to deduce from the net that X is either Tulsa or Oklahoma-City. The very ambiguity regarding which distinct element of Major-OK-Cities is actually denoted by X is attractive for some applications (as shown in the Tom, Dick, and Harry example).

Note that distinctness and disjointness properties may be propagated through a network. In particular, if A has an outgoing de arc to S1, and B has an outgoing de arc to S2, and there are unbroken paths of ds arcs from both S1 and S2 to some common superset S3, then S1 and S2 are disjoint, and A and B are distinct. In fact, every element of S1 is distinct from every element of S2.

The use of ds and de arcs increases the power of the taxonomy by making it possible to prove negative assertions. For example, with CA-Cities and OK-Cities known to be disjoint, it is possible to show that Tulsa (or X) is not a California city. Information about nonintersection and nonequivalence can be encoded by other means, but the de and ds arcs allow much of this information to be encoded for the price of the hierarchical information alone, without additional structure.

IV PARTITIONING

A. Spaces

A new dimension to the organizational and expressive power of representation networks may be added by extending the basic concept of a network as a collection of nodes and arcs to include the notion of partitioning [Hendrix 1975a, 1975b]. The central idea of partitioning is to allow groups of nodes and arcs to be bundled together into units called spaces, which are fundamental entities in partitioned networks, on the same level as nodes and arcs.*

Every node and every arc of a network belongs to (or lies in/on) one or more spaces. Associated with the data structure encoding a space is a list of all nodes and a list of all arcs that lie within the space. Likewise, associated with the data structures of nodes and arcs are lists of all the spaces upon which they lie. Nodes and arcs of different spaces may be linked, but the linkage between such entities may be thought of as passing through boundaries that partition spaces. Nodes and arcs may be created in (initially empty) spaces, may be transferred or copied (at a fraction of creation cost) from one space to another, and may be removed from a space.

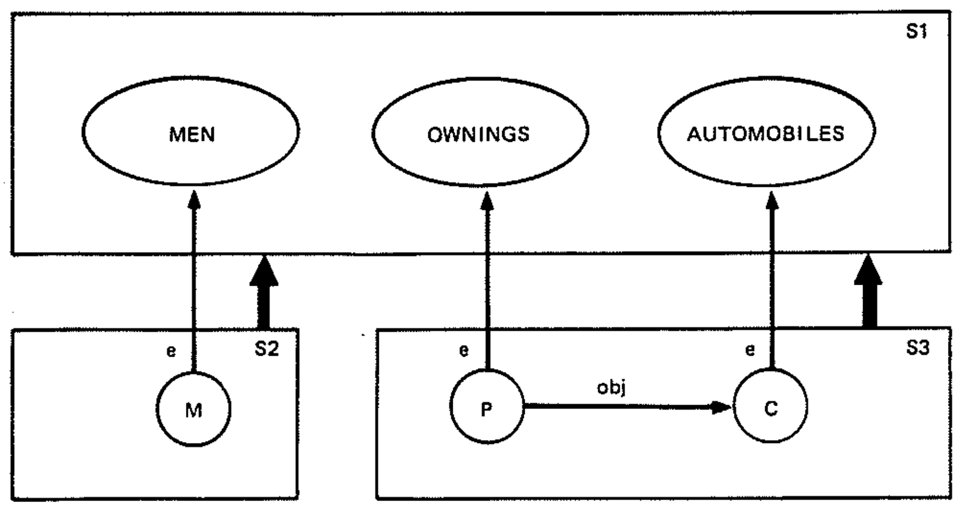


Figure 4 Partitions Around Syntactic Units

An important application of spaces in language processing, which provides a convenient introduction to the partitioning concept, is in grouping together subparts of a network that are capable of being expressed by a single syntactic unit. For example, Figure 4 shows a

----
* Aggregate structures similar to spaces have also been described by Scragg [Scragg 1975b] and Hayes [Hayes 1977], although their structures have not been applied in so wide a range of applications as have spaces.

network containing three spaces, two of which correspond to syntactic units. Each space is represented by a rectangle that contains the name of the space in a corner. Thus, space S1 is the space at the top of the figure. Diagrammatically, a node or arc is indicated as belonging to a space if its label is written within the rectangle associated with the space. So, node M and the e arc from M to MEN lie only in S2. Spaces S1, S2, and S3 may be given concrete interpretations in the context of the sentence

"SOME MAN M OWNS A CAR C".

Space S1 encodes background information (about men, owning situations, and automobiles) for the understanding of this sentence. Space S2 encodes "some man M", the information that would be conveyed by the syntactic subject of the sentence. Space S3 encodes aspects of an owning situation P in which the object owned is a car C. This corresponds approximately to the verb phrase of the sentence ("owns a car C").* Figure 4 does not in fact indicate that M was the agent in owning situation P, but this omission is corrected below.

B. Vistas

It is often convenient to combine several spaces to form a composite bundle of nodes and arcs representing the aggregate of the bundles of the individual spaces. Such a combination of spaces is called a "vista", and is somewhat like a QLISP context [Reboh and Sacerdoti 1973]. Most operations involving a partitioned network are performed from the vantage of a vista with the effect that the operations behave as if the entire network were composed solely of those nodes and arcs that lie in the spaces of the vista. All structures lying outside the vista are ignored.

The mechanics of partitioning allow vistas to be created freely from arbitrary combinations of spaces, but this freedom is seldom used. Rather, vistas are typically created in a hierarchical fashion by adding one new space to an existing vista or to the union of multiple existing

----

* Tense information is omitted.

vistas. A new vista created in this fashion inherits a view of (or access to) the information in the parent vista(s), and the newly added space is used for extending local information without altering the view of the parent(s). Such hierarchically created vistas are analogous to programming contexts with global and local variables. Information structures in the spaces of the parent vista(s) are global, relative to the new space. Because the new space S of a new hierarchically created vista V is so closely related to V, it will be convenient to talk about "viewing the net from the vantage of S" when the viewing is actually from V.

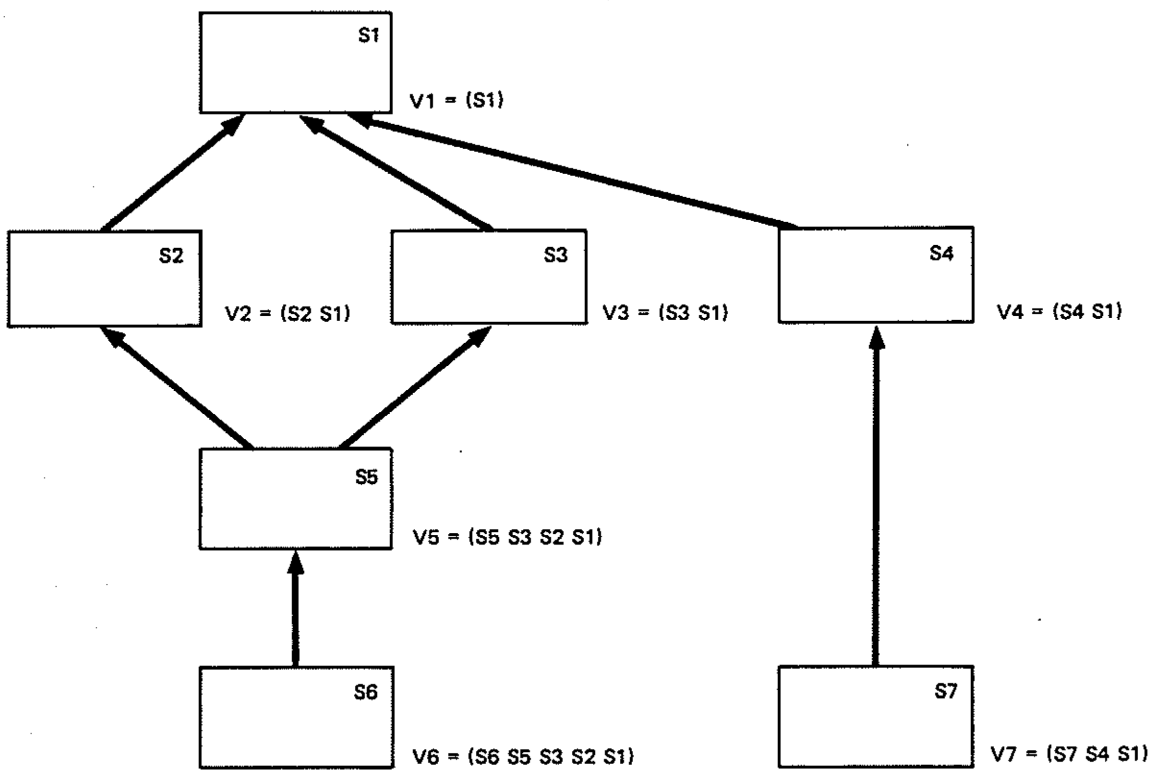


FIGURE 5 ABSTRACTION OF VISTA ORDERING

When new vistas are created hierarchically, they form a partial ordering of viewing capability. An example of such a partial ordering is depicted in Figure 5. The spaces that are included in the various vistas are represented by rectangles as before. To the right of each rectangle is a list notation (vistas are actually implemented as LISP lists) indicating the vista associated with the space. Heavy arrows indicate the inheritance of viewing capability. That is, from any point in the partial ordering, information is visible in any space that may be reached by following up heavy arrows.

Space S1 at the top of the figure is associated with vista V1, which contains only space S1. From the vantage of V1, only the information in S1 is visible. The vista of S2 is V2, which contains both S2 and S1. Thus, from the vantage of V2, all the information in both S2 and S1 is visible. However, the information in S3 is not visible from V2 (except to the extent that S1 or S2 contains some of the same nodes and arcs as S3). From the vantage of V5 it is possible to see all the information in both S2 and S3, as well as the information in S5 and S1.

Figure 6 provides some indication of how vista hierarchies may be used. Again, the heavy arrows indicate which spaces are included in the vistas of spaces. From the vista of space VP1, it is possible to see information on spaces VP1, V1, NP2, and BACKGROUND. Thus, from the vantage of VP1, it is possible to see the background information and the structures used in creating a network interpretation of the verb phrase (VP) in the sentence "Some man M owns a car C". This view includes the information of space V1 (which encodes the verb alone), space NP2 (which encodes the direct object alone), and space VP1 (which encodes the relationship between the verb and object). From this same vantage, the structures in spaces NP1 and S1 are invisible.

In subsequent diagrams, when a rectangle representing a space S is drawn completely within a rectangle representing a second space S', then this indicates that the vista of S is an extension of the vista of S'. For example, A and B in Figure 7 represent equivalent structures. If two rectangles overlap, but neither contains the other, then structures appearing in the overlap.lie on both spaces. Examples of such overlaps occur in the section on quantification below.

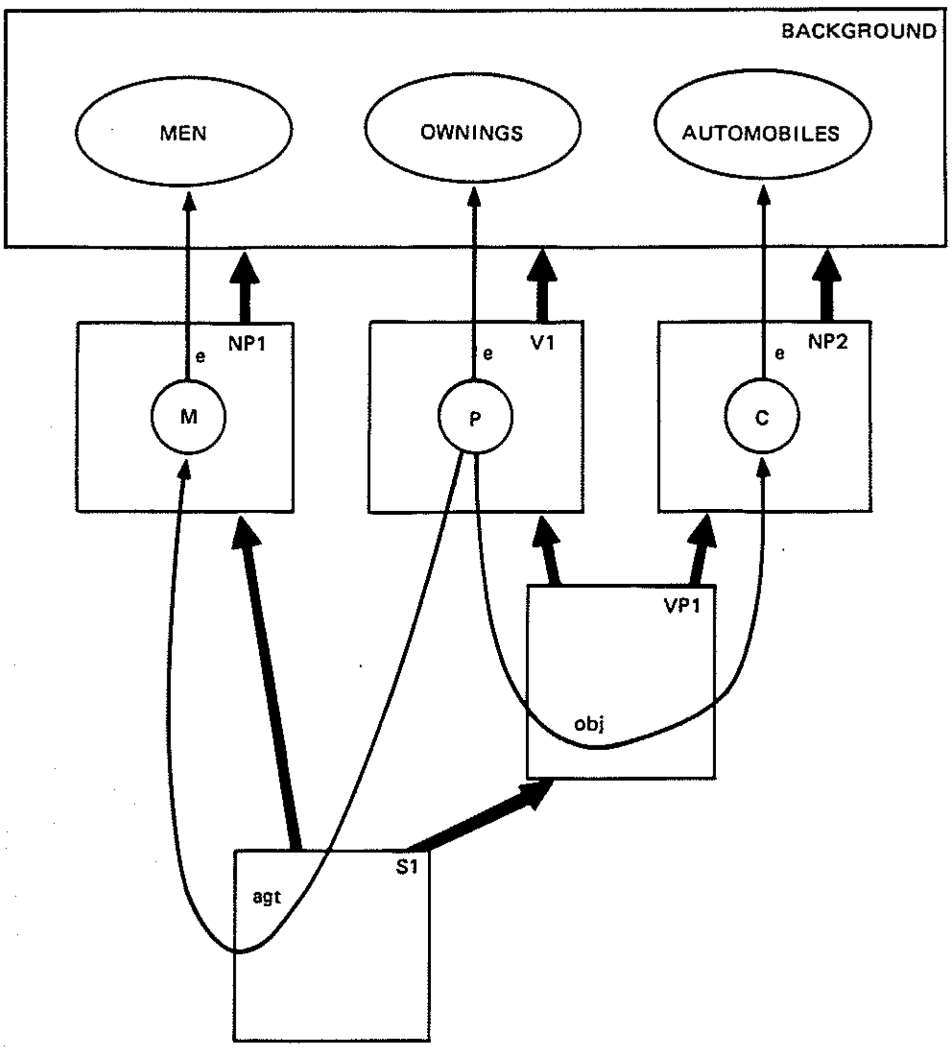


FIGURE 6 PARSING "SOME-MAN OWNS A-CAR"

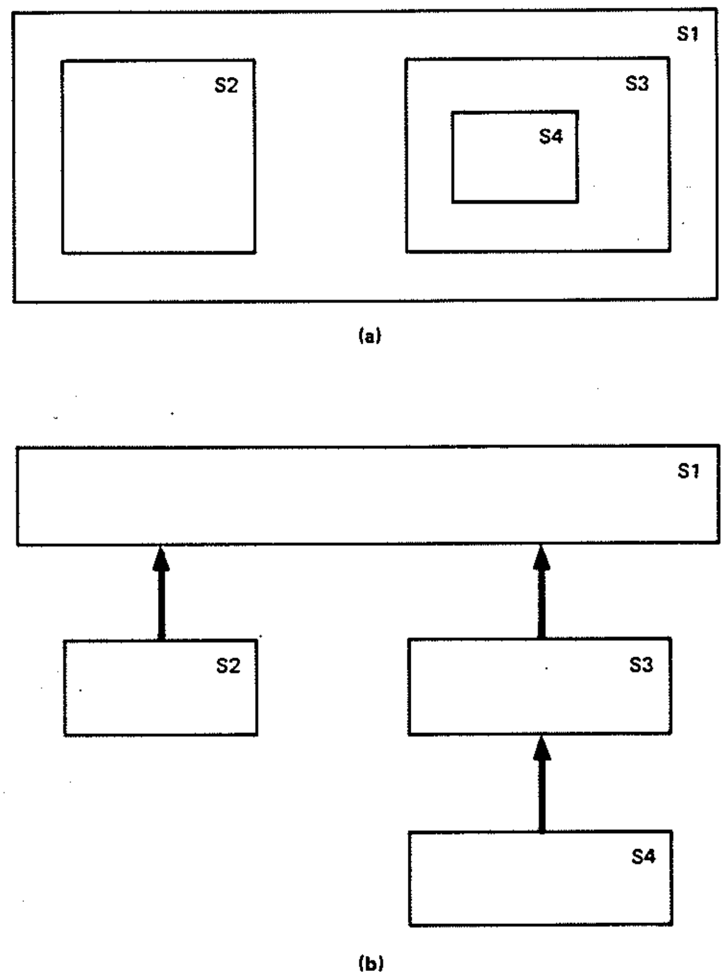


FIGURE 7 EQUIVALENCE OF ENCLOSURE AND HEAVY ARROW NOTATION

C. Relating Spaces to Predicate Calculus

The collection of nodes and arcs composing a network encodes a body of information in much the same fashion as a set of propositions in predicate calculus. If a total network is regarded as a large set of propositions, then a space may be regarded as a subset of the propositions, and a vista may be regarded as the union of a number of the subsets. Thus, for example, the vista consisting of spaces VP1, V1 and NP2 of Figure 6 contains the set of propositions that are conveyed by the verb phrase of the example sentence. This set of propositions may be thought of as a single proposition formed by conjoining the individual members of the set. Thus, the spaces and vistas may be regarded as propositions, that is, as <u>expressions</u> of information about the world. This, of course, is consistent with the notion that a network is an expression of information.

D. Supernodes

By bundling together a collection of representational structures, a space may be used as the aggregate expression of the information encoded by its internal nodes and arcs. For example, a certain space S might bundle together a collection of nodes and arcs which, when taken together, represent the set of things that some person has told about in a story, or believes to be true, or wishes to have happen. Each node and each arc represents some aspect of the belief (story, wish), but only the space is a representation of the aggregate of these aspects.

Because it is often necessary to relate other concepts in the network to the proposition encoded by a space, <u>supernodes</u> may be created to denote spaces. Supernodes have all the properties of ordinary nodes, and in particular may be pointed to by arcs.

When a supernode is formed to denote a space, a QUOTE-type operation takes place. The supernode comes to denote the <u>expression</u> of the information represented by the space. That is, the space represents information about the modeled domain, and the node denotes the representation.

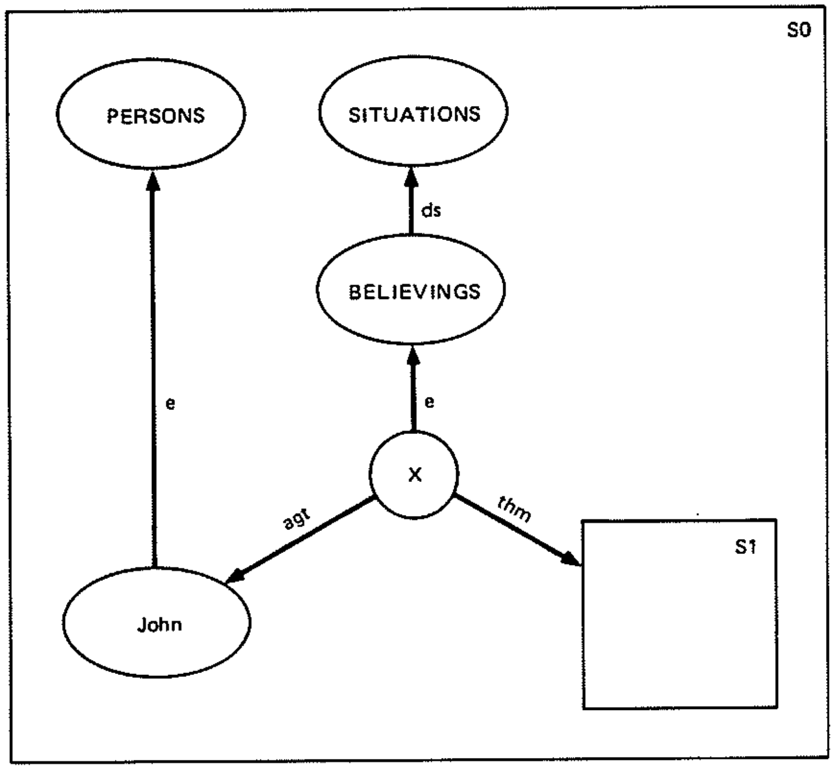


FIGURE 8 THE BELIEFS OF JOHN

An example supernode is shown abstractly in Figure 8. Node X represents a believing situation in which the believer (agt = agent) is JOHN and the thing believed (thm = theme) is a complex of information encoded by space S1. More precisely, the structures inside S1 (omitted in the figure) may be thought of collectively as a complex proposition that JOHN believes to be true. Moreover, the structures of the space represent objects and situations that JOHN believes to exist.*

It is important to note that by allowing the network to express information about expressions, the use of supernodes can lead to interesting inconsistencies and paradoxes, some of which are discussed by Montague [Montague 1974]. For example, the Liar Paradox ("this statement is not true") is easily expressed in a partitioned network.

----

* A discussion of the use of partitioned networks in modeling belief structures is presented in [Cohen and Perrault 1976].

Fortunately, these problems have had little impact on our current work. This is because our systems that manipulate information about expressions currently limit their scope of activity to information about the standard logical connectives. For example, these systems know that if the disjunction of expressions E1 and E2 provides an ambiguous description of some aspect of a world W, then either the objects and situations described by E1 or those described by E2 exist in W.

V STRUCTURES FOR LOGICAL DEDUCTION

Building upon the basic notions of nodes, arcs, spaces, vistas, taxonomies, situations, and deep cases, structures may be devised to meet the needs of various applications. This section describes how the basic notions may be extended to meet the needs of a system that does logical deduction. In particular, structures are described for handling logical connectives and quantification.

A. Logical Connectives

1. Conjunction

As the first logical connective, consider conjunction, which relates a number of components called "conjuncts". Thinking of each conjunct as a description of some condition, the conjunction itself is a complex description of the situation in which the conditions described by each of the individual conjuncts exist in unison.

The inherent bundling capability of spaces makes them a convenient medium for dealing with conjunction. In particular, a conjunction C corresponds to a space S upon which network structures are created corresponding to each conjunct of C (and only the conjuncts of C). Space S2 of Figure 9, for example, represents the information conveyed by the conjunction "Old.Black was built by Ford <u>and</u> Old.Black is owned by John". The subordination of S2 under S1 in the viewing hierarchy is rather artificial and was done here solely for exposition.

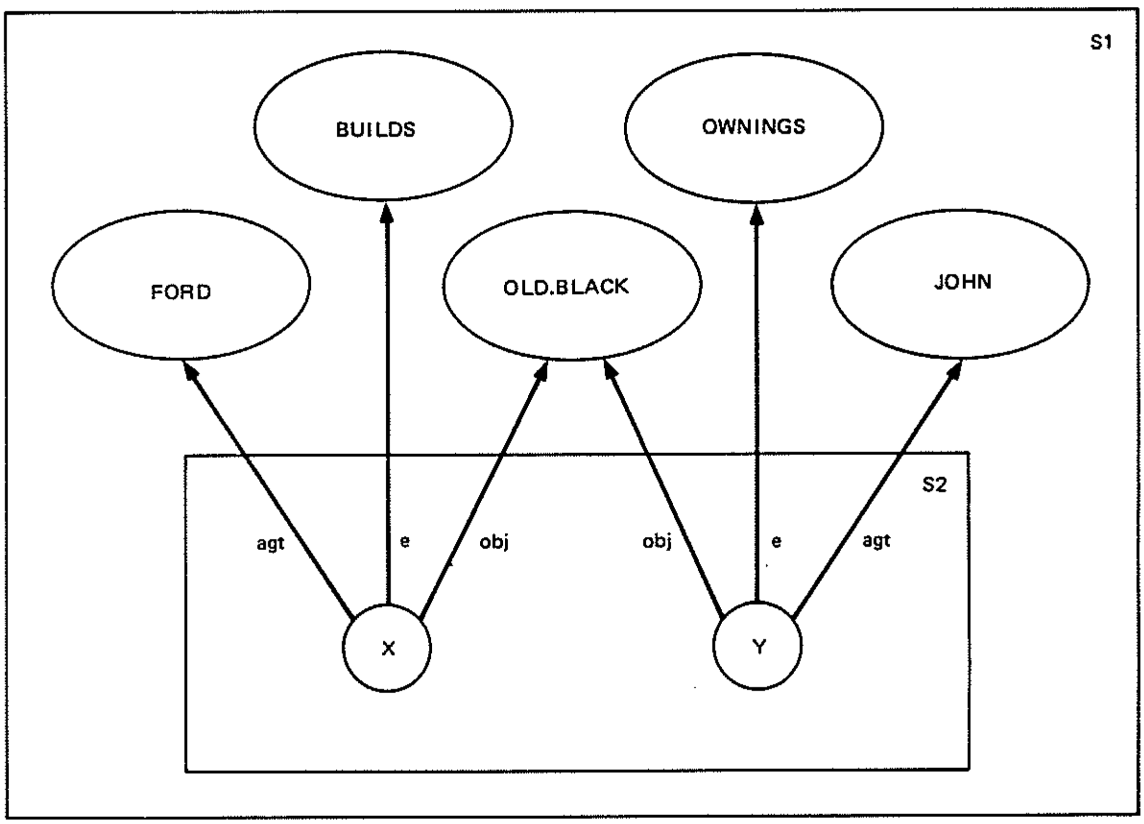


FIGURE 9 OLD.BLACK WAS BUILT BY FORD AND IS OWNED BY JOHN

Except for delimiting the conjunction (X & Y), the structures of S2 might just as well have been encoded directly in S1. This ability to remove the partitioning of S2 is the network analog of the ability to remove the embedded parentheses in the formula (A & (B & C)) to form (A & B & C).

2. Disjunction

A disjunction separates out a number of components called "disjuncts", each of which describes an alternative set of conditions. The inherent separating ability of spaces makes them a convenient medium for dealing with disjunction. In particular, the information encoded by each of the n disjuncts of a disjunction D may be encoded on a different space and so kept (and reasoned about) in (relative) isolation.

Figure 10, for example, shows the network encoding of the disjunction D = "Either Old.Black was built by G.M., or Old.Black is owned by John". Node D denotes the disjunction itself. It is an element of the set Disjunctions, which might more properly be labeled "True Disjunctions" because it denotes the set of sets of propositions in which at least one proposition represents entities that exist in the modeled world. That is, a disjunction like D can belong to Disjunctions in some world W only if the objects and situations described by at least one of its disjuncts exist in W. The disjuncts of D are represented by supernodes S2 and S3. Since a disjunction may be regarded as a set of alternative disjuncts, the disjuncts of D are shown as distinct elements of D. Whenever a disjunction appears in the network, it is assumed that all members of the disjunctive set are explicitly encoded.

The entire disjunction structure is embedded in the conjunction of S1. S1 provides a partial description of some world (i.e., a collection of objects and the interrelationships among them) and each structure in S1 represents some object or situation that occurs in that world. So, when the network is viewed from the vantage of S1, such entities as Old.Black and D are seen to occur. However, the structures in spaces S2 and S3 are not seen from the vantage of S1 and

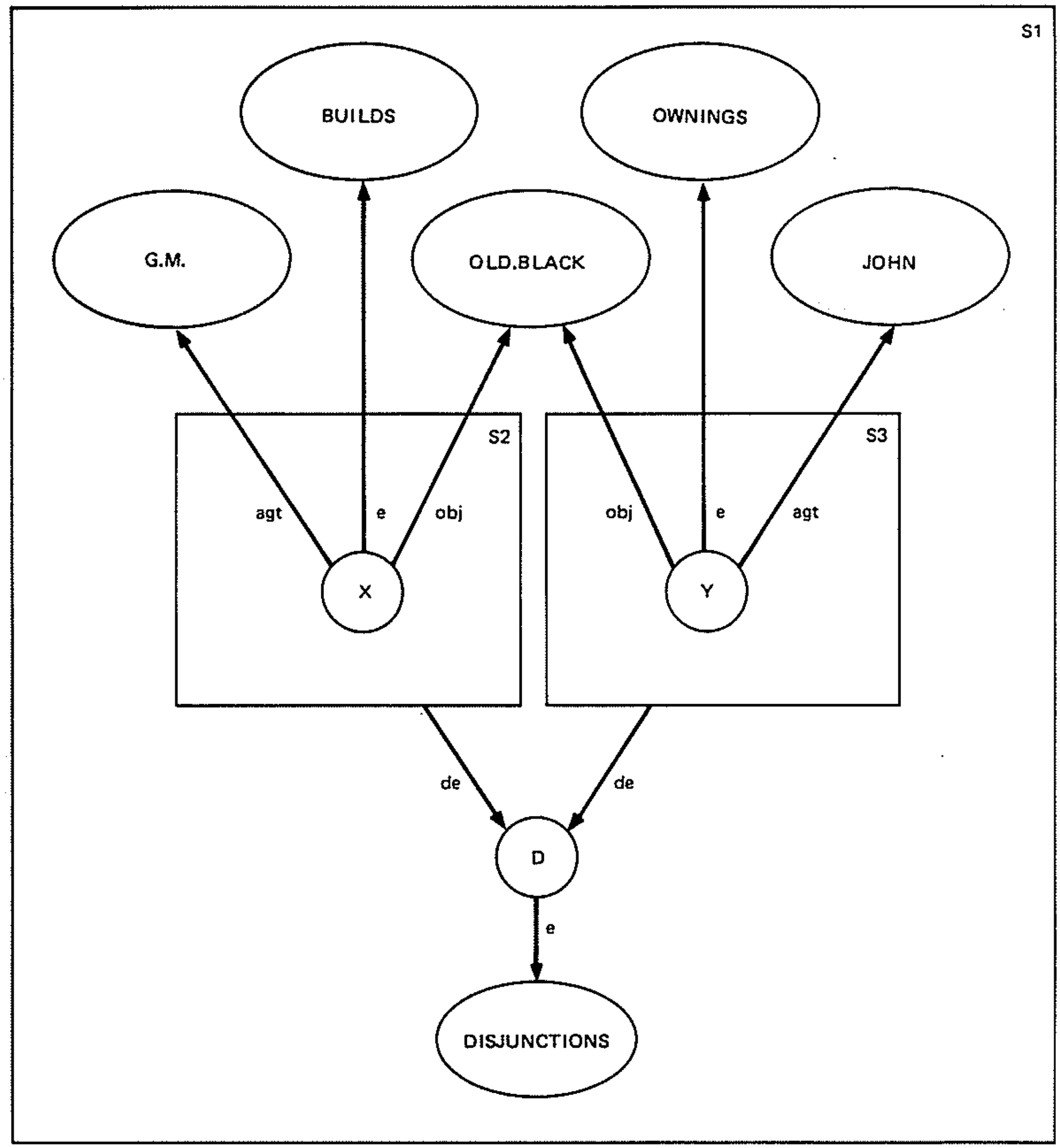


FIGURE 10 EITHER OLD.BLACK WAS BUILT BY G.M. OR OLD.BLACK IS OWNED BY JOHN

are thus not asserted in the world modeled by S1. Since D does appear in the world of S1, it is known that the world of S1 includes the situations described by at least one of the disjuncts of D. If S2, for example, were included, then the modeled world would include all situations described by structures that are visible from the vantage of S2. This view includes structures in S2 and S1, but excludes structures in S3.

3. Negation

The network encoding of negation uses partitioning to separate the negative from the positive. Figure 11 shows the network encoding of the negation "G.M. did not build Old.Black". The negation, an element of NEGATIONS*, is encoded by supernode S2. Space S2 is an (implicit) conjunction describing a set of conditions that cannot occur simultaneously in the context of the conditions described in S1. As in the disjunction example, the negated structures inside S2 are not visible when viewing the network from the vantage of S1, although the negation itself is visible. That is, the description (denoted by supernode S2) of the negative conditions (described by space S2) exists in the world of S1, and the fact (denoted by the e arc from S2 to NEGATIONS) exists in S1 that this description encodes a nonexistent set of conditions for the world of S1. But entities described by S2 do not (all) exist in S1.

4. Implications

Implications can be encoded by conversion to negations and disjunctions. (A => C is the same as ~A v C in propositional logic.) However, it is also possible to encode implications directly, as shown abstractly in Figure 12. An implication I occurring in space S is associated by case arcs with a collection of antecedent conditions, represented by a space A, and a set of consequent conditions, represented by a space C. As above, spaces A and C may be regarded as conjunctions.

---

* In a world W, Negations is the set of propositions that are not true in W.

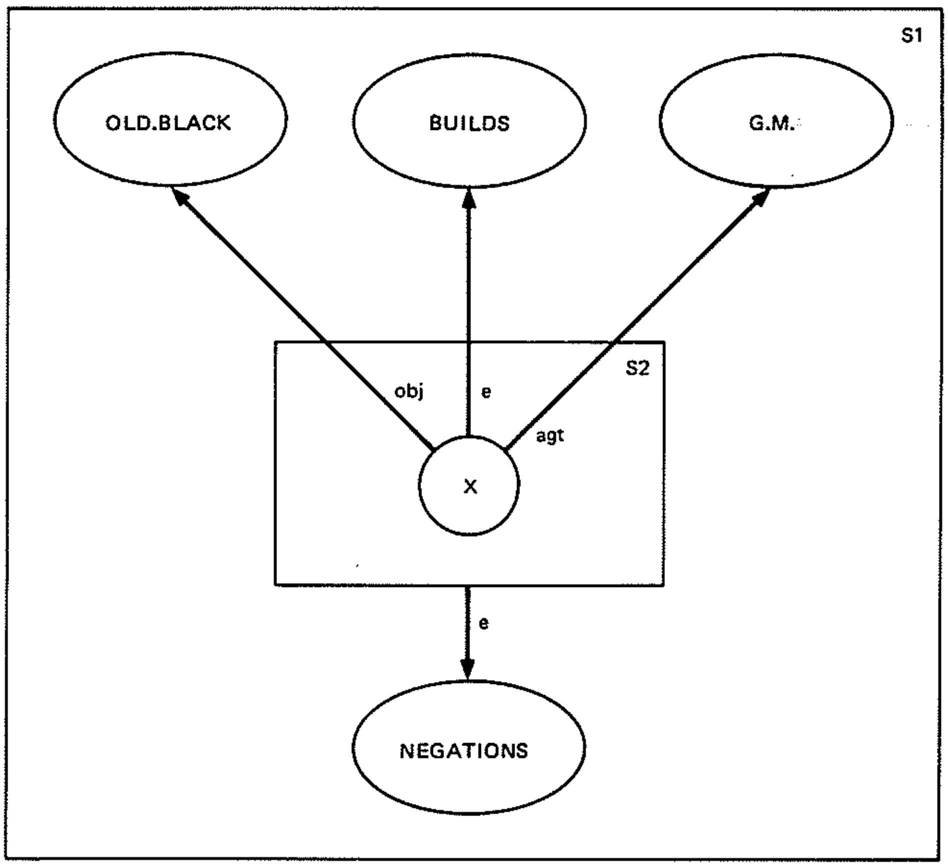


FIGURE 11 OLD.BLACK WAS NOT BUILT BY G.M.

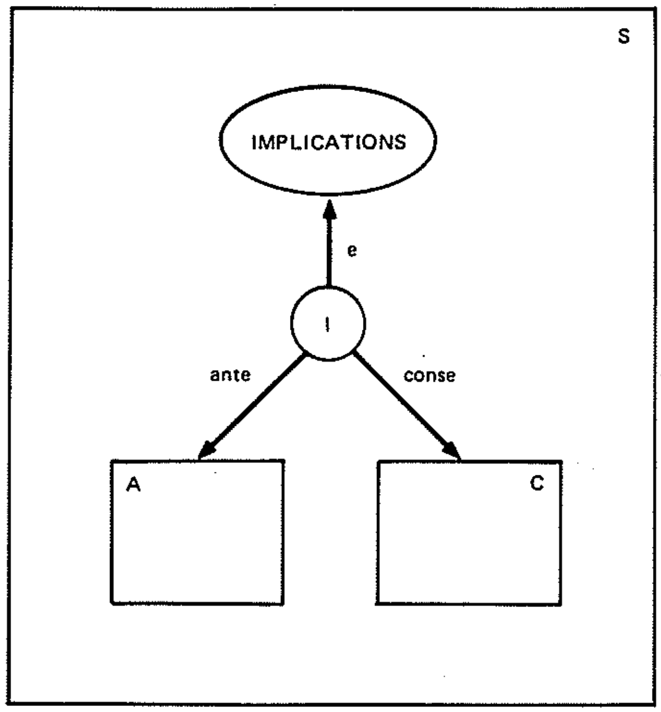


FIGURE 12 ABSTRACTION OF AN IMPLICATION

B. Quantification

One of the important features of network partitioning is that it provides a facility for representing arbitrarily nested existential and universal quantifiers. Existential quantification is an implicit concept in the sense that the occurrence of a structure (i.e., a node or arc) in a space is taken to be an assertion of the existence with respect to that space of the entity represented by the structure. Existential quantification and negation could be used to represent any universally quantified formula* (Ax in X)P(x) by making use of the following transformation:

(Ax in X)P(x) <=> ~~[(Ax in X)P(x)] <=> ~[(Ex in X)~P(x)].

Although the network encoding suggested by this transformation is

----

* "A" is written for universal quantification, "E" for existential quantification, "in" for set membership, and "<=>" for equivalence.

logically sound, it is cumbersome and unappealing intuitively. The following transformation suggests a more attractive representation:

(Ax in X)P(x) <=> (Ax)[member(x, X) => P(x)].

That is, any universally quantified formula can be represented as an implication whose antecedent specifies the <u>typing</u> of the universally quantified variable and whose consequent specifies the statement that is being made about any entity that satisfies the type restrictions.

A distinguishing feature of the universally quantified variable x in this implication is that it occurs in both the antecedent and the consequent. Thus, in the network representation of an implication, if a node occurs in both the antecedent and the consequent space, it is considered to represent a universally quantified variable*.

Figure 2 shows the representation of a concrete example of such an implication, namely the statement "For all M in the set of Mustangs, there exists a B such that B is an element of the set of Builds situations, the agent of B is Ford, and the object built is M", or

Am[member(m, Mustangs) => built(Ford, m)]

When the main connective of a formula is an implication, it is not necessary to embed the formula in another implication to represent the universal quantification. That is:

```
        (Ax in X)[Q(x) => R(x)]
<=>     (Ax){member(x, X) => [Q(x) => R(x)]}
<=>     (Ax){[member(x, X) & Q(x)] => R(x)}.
```

Arbitrary nesting of quantifiers may be achieved by placing implications in the consequent spaces of other implications. For example:

```
(Ax in X)(Ey in Y)(Az in Z)P(x,y,z)   <=>

(Ax){member(x, X) =>
      (Ey)[member(y, Y) & (Az)(member(z, Z) => P(x,y,z))]}.
```

---

* Note that the constant K in Ax[P(x, K) => Q(x, K)] does not <u>appear</u> in both ante and conse spaces, but is <u>referenced</u> (i.e., pointed to by an arc) in both ante and conse. For example, FORD is referenced in the conse space of the implication of Figure 2.

C. A Deduction Algorithm

The structures presented above are useful in deduction and question answering only to the extent that there exist procedures having the logical expertise needed to manipulate them. The general flavor of such procedures, which are discussed more extensively in [Fikes and Hendrix 1977], is indicated by the following, highly simplified example.

The deduction system is given as input a QUERY space representing a question to be answered (theorem to be proved) and a KNOWLEDGE space representing the beliefs that are to be considered true while answering the question. In aggregate, the nodes and arcs of QUERY describe a set of objects and relationships whose existence is to be established in the world of the KNOWLEDGE space. If a set of such entities can be found, a list of bindings that link the QUERY descriptions to their KNOWLEDGE instantiations is to be returned.

For example, Figure 2 shows a QUERY and KNOWLEDGE space for the question "What company built Old.Black?" Given this problem, the system seeks an element (like Z) of the Builds situation set having both Old.Black as its object and an element (like ?X) of the Companies set as its agent. Looking for a match for Z, the system first looks only at structures in the KNOWLEDGE space. The Builds situation represented by node P is found by using the incoming e arcs to the BUILDS node as an index. However, P is rejected as a match for Z because the obj arcs from Z and P point to nodes that are members of disjoint sets, indicating that Z and P have different objects.

Because there are no other incoming e arcs to BUILDS in the KNOWLEDGE space, the system changes strategy and looks for elements of the Builds set that are buried in logical expressions. Indexing again on the incoming e arcs to the BUILDS node, the "theorem" "All Mustangs were built by Ford" is found. A unification process determines that the relevant instance of the theorem is one in which the universally quantified variable M is instantiated by Old.Black. The theorem allows a new Builds situation to be asserted if it can be shown that Old.Black is an element of the Mustangs set. A subproblem is created to find that

ElementOf relationship, and when the subproblem is solved, a new Builds situation is asserted in KNOWLEDGE and the desired bindings are assigned. In particular, node ?X is bound to FORD and Z is bound to the newly derived Builds situation. Because node ?X is marked for questioning, its binding, FORD, is returned as the answer to the original question.

VI INHERITING INFORMATION

A. The Contributions of Quantification and Case

The notion of a hierarchical taxonomy was presented earlier as a basic concept in nets used for the representation of knowledge. Although the taxonomy alone provides the information needed to answer several types of element/set/subset questions, the taxonomy's primary attraction is in supporting the "inheritance" of information. In particular, if certain properties P are known to be characteristic of all the members of a given set S, then it follows that all the members of the set's subsets also have the properties P and that each individual member of the set S and of its subsets has the properties as well. To explicitly reencode the properties P with each of the individual representations for S, its subsets, and its many individual members would be highly redundant. Instead, the properties P may be recorded solely with the representation of S and deduced for the subsets and individual members on demand. For example, if the fact that dogs have cold noses is recorded with the node representing the set of all dogs, then it is unnecessary to explicitly encode the fact that all members of the set of Hounds and the individual dog Fido have cold noses.

The procedures implementing deductions based on set membership and set inclusion may make very efficient use of the network data structures that encode taxonomic information. For example*, in looking for a

---

* In this example, given particular object Q we ask if Q has property P. A more demanding application of the inheritance information, which is discussed in [Moore 1975] and in [Fahlman 1977], is to find an object Q that has property P.

property P of an object Q, these procedures may determine whether P is recorded with the data structure representing Q. If it is, no deduction need be done. If it is not, the procedures may work their way up the taxonomy above Q, checking to see if information about P is encoded with any of the sets and supersets to which Q belongs. Because such a search may be efficiently implemented, the deduction procedures make it appear as if the representation of object Q had inherited its own copy of information about P from supersets in the taxonomy.

Unfortunately, many network schemes have been devised that indicate inheritance either incorrectly or through a complicated system of ad hoc rules and structures. Various shortcomings are catalogued in [Woods 1975] and in [Brachman 1977]. Those systems that behave incorrectly suffer primarily from the failure to carefully distinguish properties of sets (such as cardinality) from properties of individual members of a set. For example, the individual members of a set of numbers may all be prime, but the set is not prime. Those systems that are clumsy or ad hoc fail to recognize that the encoding of information regarding all the individual members of a set inherently involves universal quantification.

In general, if a set S has some property Q, it should be encoded by the network formulation of Q(S). Q might be encoded by a single arc or by a situation node with a case arc to S and other arcs to other participants in the situation coded by Q. If the individual members x of S all have property P, then this information should be encoded by the network formulation of Ax[member(x, S) => P(x)]. Note particularly that P is not applied to S but to the values of the universal variable x that ranges over S.

Along with quantification, the notion of case plays an important role in "inheritance". Our explicit restriction that case arcs designate only instances of functions is a key (but often unrecognized) factor in network deduction. Suppose, for example, that it is known that

```
Ax{member(x, Ownings) => Et1,t2 [start-time(x, t1) &
                                 end-time(x, t2)  &
                                 BEFORE(t1, t2)  ]}
```

& member(q, Ownings)
& start-time(q, I)
& end-time(q, J)

From this information, is it possible to deduce that I, the start-time of q, is BEFORE J, the end-time of q? Certainly, the universally quantified statement (UQS) indicates that there exist T1 and T2 which are <u>a</u> start-time and <u>an</u> end-time for q with T1 BEFORE T2, but it is the fact that start-time and end-time are cases, and therefore functions, that indicates that T1 and T2 are necessarily identical to I and J respectively.

In general, for any case relation c, the function property brings with it the restriction that

$$Ax,y,z\{[c(x, y) \ \& \ c(x, z)] \Rightarrow y=z\}$$

The point here is that for a composite object (some object associated with cases), not only may the object itself "inherit" properties, but the associated objects that fill case roles may also "inherit" properties. That is, from a UQS of the form

$$Ax\{member(x, S) \Rightarrow P(x) \ \& \ Ey[c(x, y) \ \& \ Q(y)]\}$$

where c designates a case relation, it follows that

If A is in S, then P(A)

and

If c(A, B)[*], then Q(B).

B. <u>Delineation</u>

By indicating some common property P of members of a set S, a universally quantified statement serves to partially define and bound S. That is, by stating that all members of S have property P, a UQS indicates that <u>only</u> individuals having property P are in S. Thus, the UQS provides an indication of a limitation on the membership of S. Formally, this limitation arises as a consequence of the fact that

$$\{Ax[member(x,S) \Rightarrow P(x)]\} \ \Leftrightarrow \ \{Ax[\sim P(x) \Rightarrow \sim member(x,S)]\} \ .$$

----

[*] That is, if B fills the role c for A.

For purposes of understanding natural language inputs, UQSs serving to limit the membership of situation sets are very important. In particular, it is useful for each situation set to have a UQS, called the set <u>delineation</u>, that names and restricts the participants of situations in the set. That is, the UQS specifies deep cases that are to be associated with situations of the type being delineated and indicates a possible set of values for each case. For example, the delineation of the set Ownings is shown in Figure 13, and corresponds to the formula

```
Ax{member(x, Owings) =>
     Ey,z,t1,t2[member(y, Legal.persons) & agt(x, y) &
                member(z, Physobjs)  &   obj(x, z) &
                member(t1, Times) & start-time(x, t1) &
                member(t2, Times) &  end-time(x, t2) ] }
```

This UQS indicates that all Ownings situations have an agt, obj, start-time, and end-time. Further, the agt must be a member of Legal.persons, the obj must be (in this system) a member of Physobjs, and the start-time and end-time must be elements of Times. More complex restrictions may also be added. For example, the start-time could be restricted to precede the end-time, and the obj could be excluded from the set of Humans.*

By using delineations, a speech or natural language understanding system is able to reject certain anomalous combinations of phrases that nevertheless meet syntactic and acoustic criteria for being joined. For example, if various indicators suggest that an input utterance mentions an ownership situation in which the role of agt (not obj) is played by an automobile, then the delineation of OWNINGS may be used to reject the hypothesis on the grounds that the role of agt may be filled only by elements of Legal-Persons.

----

* The information that may be encoded by delineations appears to cover the information that is encoded on what Brachman [Brachman 1978] calls the "epistemological" level of descriptions.

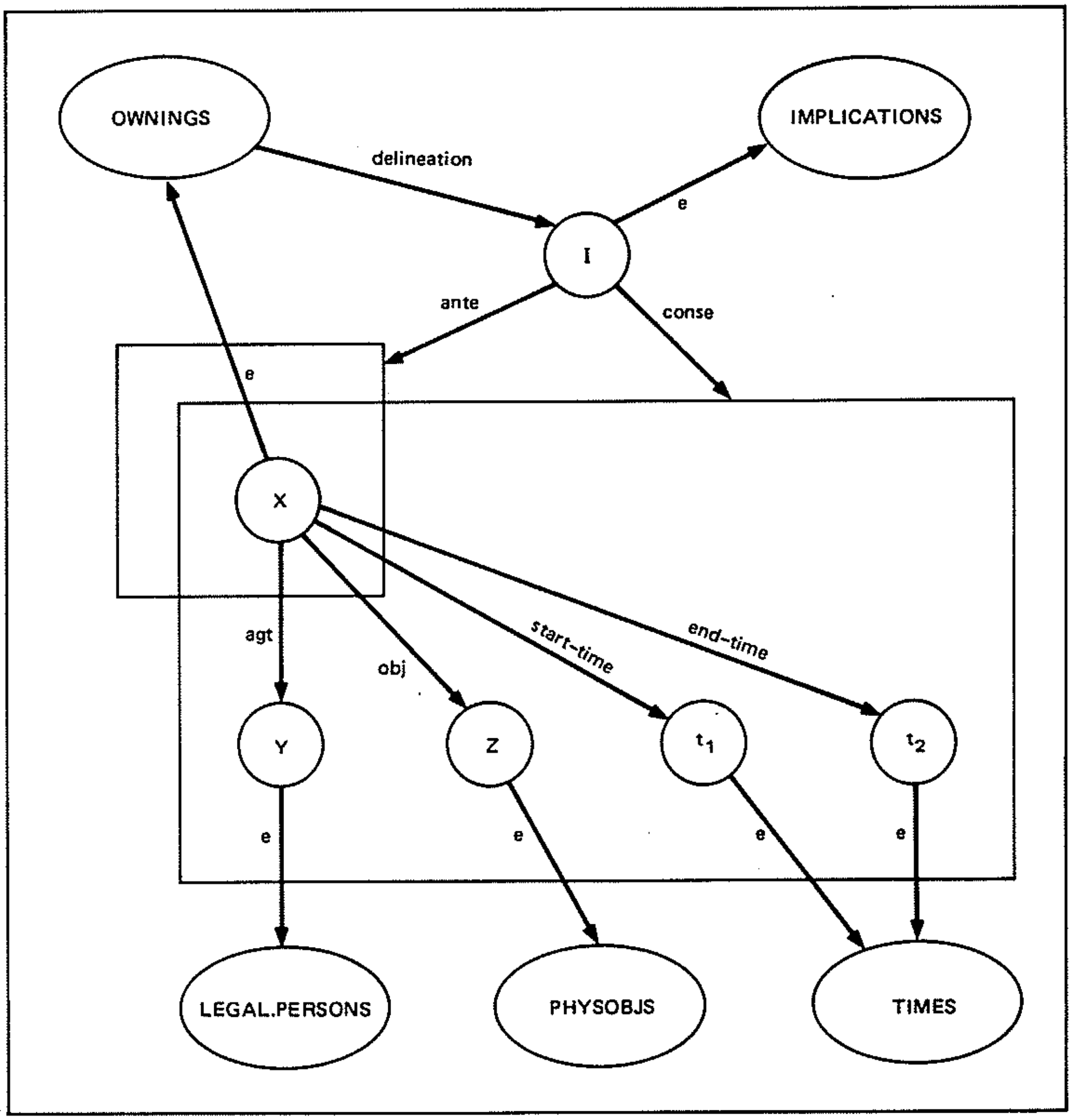


FIGURE 13 THE DELINEATION THEOREM OF OWNINGS

VII STRUCTURES FOR JUDGMENTAL REASONING

There are a number of reasoning tasks that require an ability to deal with sketchy and/or uncertain information. For these tasks, the precise rules and two-valued logic of conventional deduction systems are too confining. However, such systems as MYCIN [Shortliffe 1976] and PROSPECTOR [Duda, Hart and Reboh 1977] have dealt successfully with uncertain reasoning by using judgmental production rules. For example, PROSPECTOR uses such rules as

"Limonite casts suggest the probable presence of pyrite".

Such rules resemble an implication

$$E1 \,\&\, E2 \,\&\, \ldots \,\&En \Rightarrow H$$

here the Ei are individual pieces of evidence suggestive of a hypothesis H. Although the presence of evidence Ei seldom implies H with certainty, it is usually possible to give some rough estimate of both the necessity and the sufficiency with which some condition Ei indicates the existence of H.

Judgmental rules of this type may be encoded in partitioned networks by generalizing the structures used to encode logical implication. In particular, the Ei of a judgmental rule are placed in the implication's antecedent, the hypothesis is placed in the consequent, and the implication node is given two new case arcs indicating the necessity and sufficiency of the evidence in support of the hypothesis. (PROSPECTOR stores necessity and sufficiency on the node property list, rather than use the more expensive case arcs.)

Many judgmental rules are needed to produce a functional system. Moreover, because the hypothesis of one rule may be evidence for another, the antecedent and consequent spaces are effectively chained together into what may be called an <u>inference net</u>. When incoming information changes the probability of any piece of evidence, the probabilistic implications will ripple through the net, reassigning the probabilities of many hypotheses.

The use of partitioned networks to build inference nets and the processes for propagating probabilities are discussed in [Duda, Hart, Nilsson and Sutherland 1977].

VIII STRUCTURES FOR REASONING ABOUT PROCESSES

Systems, such as those described in [Fikes, Hart and Nilsson 1972], [Hendrix 1973], [Scragg 1975a] and [Sacerdoti 1977], that do planning or other reasoning about processes have made extensive use of state-of-the-world models (SWMs) and operators that describe how one state may be transformed into another. Partitioned networks offer attractive structures for encoding both SWMs and operators.

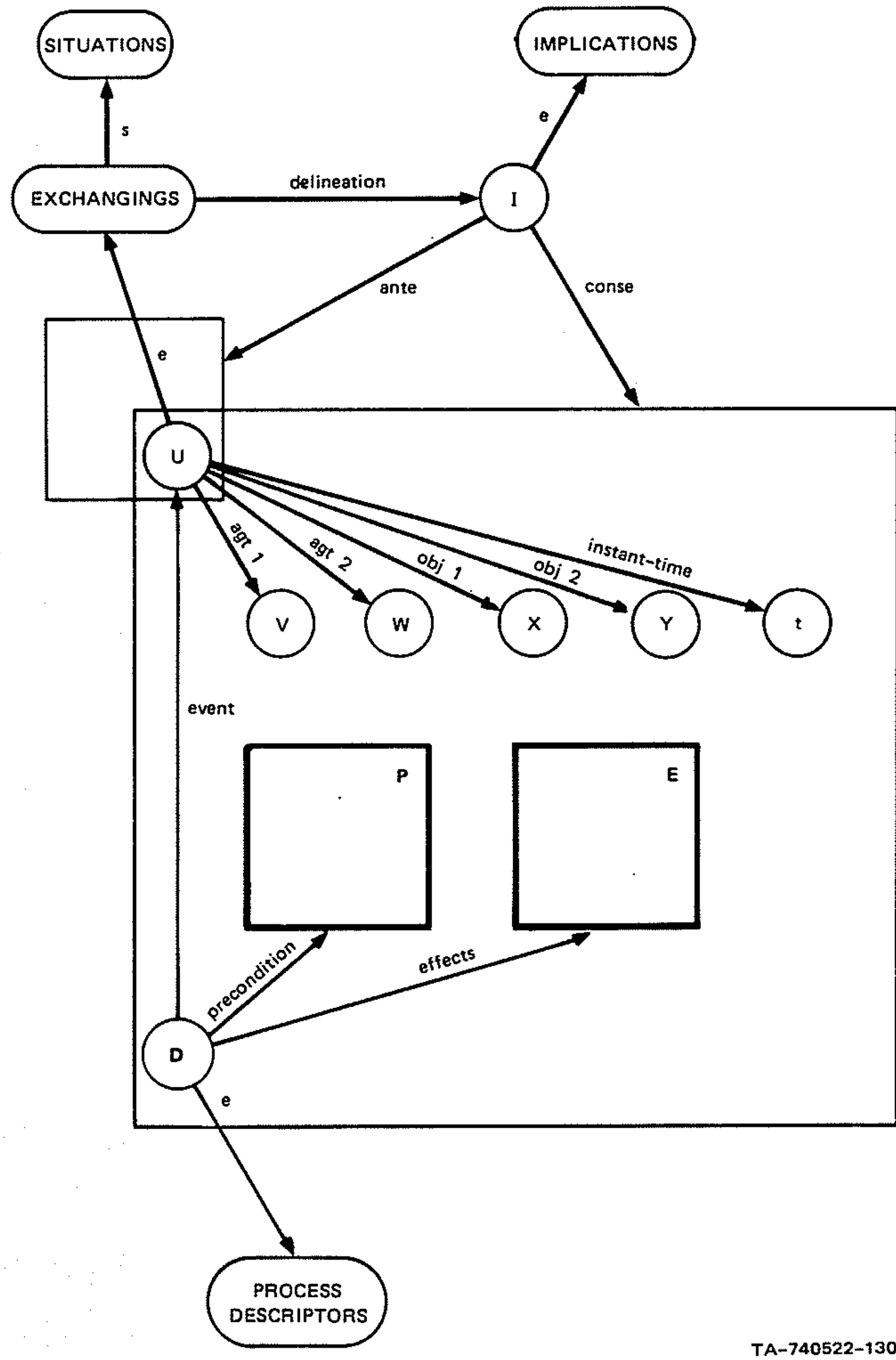
Hopefully, it is clear that a given world state can be modeled by a space containing structures representing the various conditions existing in the given state*. Relationships between states may be encoded by structures that point to the state descriptions as supernodes. To the extent that two states are similar, their encoding spaces may share common nodes and arcs.

Operators also may be encoded conveniently in networks, and are, in fact, needed to express much of the meaning of event situations. For example, Figure 14 shows an abstraction of the delineation of the set Exchangings. As in the delineation of other situations, the various deep cases associated with an exchange are indicated. The delineation also contains an event descriptor D which aids in encoding some of the dynamic aspects of an exchange. In particular, D indicates that any exchange U will be associated with certain preconditions P and certain effects E.

Both P and E are conjunctions of conditions that may reference the process parameters v, w, x, y, and t. For an instance of Exchange to exist with particular bindings for the variables, the preconditions must

----

* States of the world are represented in such systems as STRIPS [Fikes, Hart and Nilsson 1972] by sets of propositions, which are the predicate calculus analog of spaces.



TA-740522-130R

FIGURE 14 A COARSE DESCRIPTION OF EXCHANGE PROCESSES be met with the same bindings. If the exchanging does occur, then all of the effects (which may include disjunctions to represent uncertain outcomes) are implied just as if the implication

$$Av,w,x,y,t[exchange(v,w,x,y,t) \Rightarrow effects(v,w,x,y,t)]$$

had been encoded explicitly.

An important aspect of processes is that they usually may be decomposed into a sequence of subprocesses. The delineation of Shopping events, shown abstractly in Figure 15, has provisions for such a decomposition. Preconditions and effects may be used to understand shopping at a course level of detail. For a finer look, the event discriptor includes a <u>plot</u> that shows how shopping decomposes into a sequence of subprocesses. This plot takes the form of a transition network, which is similar to the ATNs [Woods 1970] used for parsing sentences. Rather than parse or generate sentences in a language, this transition net may be used to recognize or generate sequences of events that in aggregate constitute a shopping event. For example, one successful path through the <u>event grammar</u> is: GO-STORE-1, GO-STORE-2, EXCHANGE (at STORE-2), GO-HOME.

Each node in the plot network is actually a variable indicating an element of some event set. Since each event set has its own delineation, the subevents may be understood either in terms of their preconditions and effects, or recursively, in terms of their own plots (when available). For example, the instance of Exchangings in Figure 15 may be expanded through the delineation of Exchangings in Figure 14.

IX STRUCTURES FOR NATURAL LANGUAGE UNDERSTANDING

This section discusses the use of networks in a natural language understanding system (NLUS).

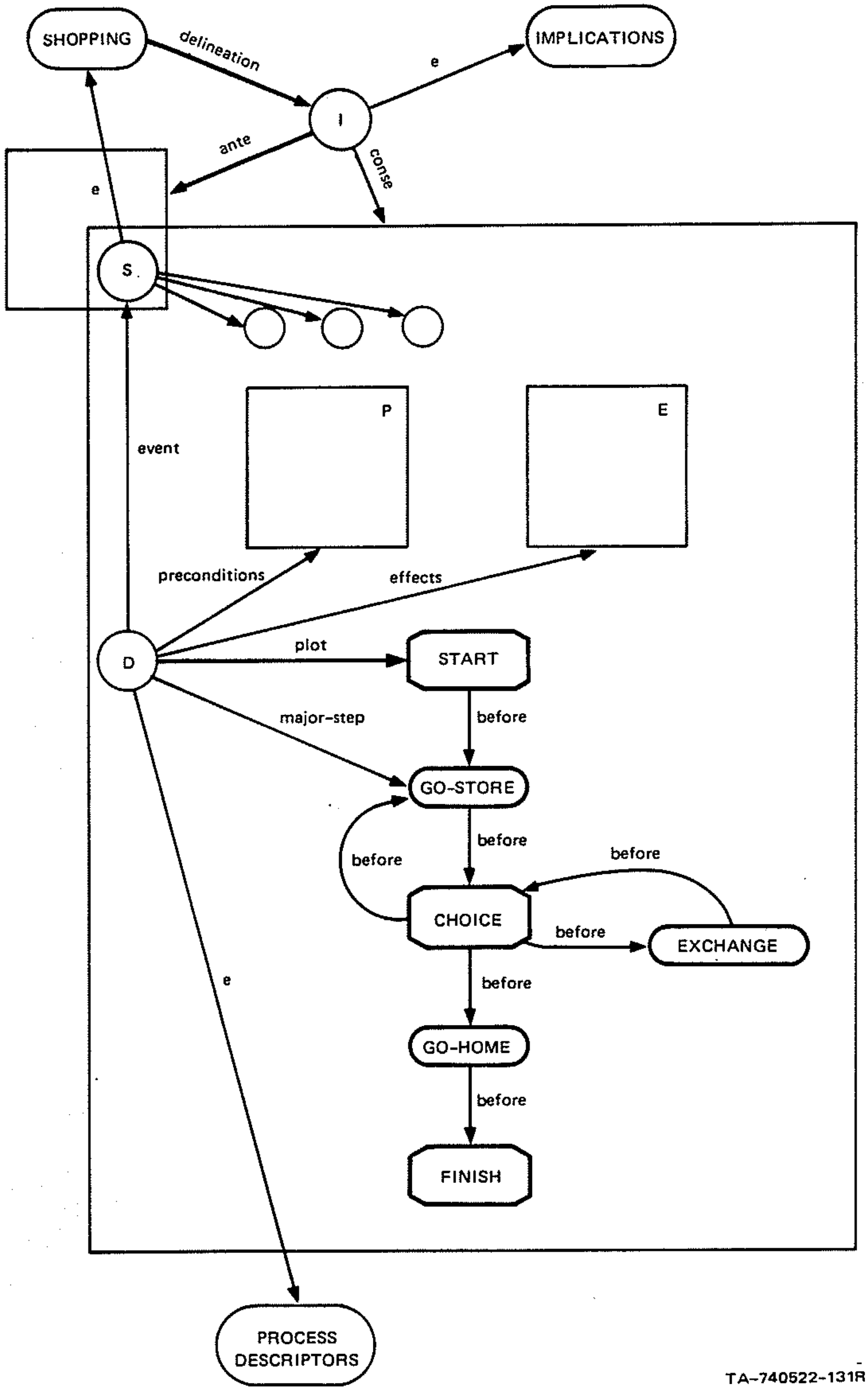


FIGURE 15 PLOT FOR SHOPPING

A. A Simple Example

To introduce the most basic aspects of using nets in a NLUS, consider the translation of the sentence

"SOME MAN OWNS A CAR."

The ultimate result of the literal translation process for this sentence is the network structure in space SCRATCH of Figure 16. Structures representing inputs are constructed in scratch spaces to separate them from the system's model of the domain of discourse, which is recorded on a space labeled BACKGROUND. (Only a fraction of BACKGROUND appears in the figure.)

The interpretation of SCRATCH space structures is quite simple: Node M denotes some man (an element of Men), C denotes some Automobile, and P denotes an Ownings situation in which the agent is M and the object is C. Because the input is understood through its relationship to previous knowledge, several arcs in the SCRATCH space link the interpretation of the new input to BACKGROUND anchor nodes. Also, because the SCRATCH space is meaningful only in the context of BACKGROUND, the vista of SCRATCH includes BACKGROUND.

As a supplementary feature, the figure also includes a node S that represents an element of the set of Meaning Situations. In particular, S associates the linguistic entity "some-man-owns-a-car" (i.e., the sentence itself) with its interpretation. Thus, the net encodes the semantics of the sentence in the conventional sense of "semantics" as a relationship between linguistic objects and their meanings*.

To suppress syntactic technicalities, assume the highly simplified language definition:

GRAMMAR	LEXICON
R1: S => NP VP	NP: some-man, a-car
R2: VP => V NP	V: owns

* The proposition denoted by the supernode SCRATCH captures only a small (but very central) part of the total meaning of the sentence. The proposition refers to concepts such as the set of Men and the set of Ownings. Clearly, the total meaning of the sentence must take into consideration what it means to belong to these sets. Moreover, a large number of inferences may be drawn from the information recorded on SCRATCH. These inferences, too, are a part of the extended meaning of the sentence.

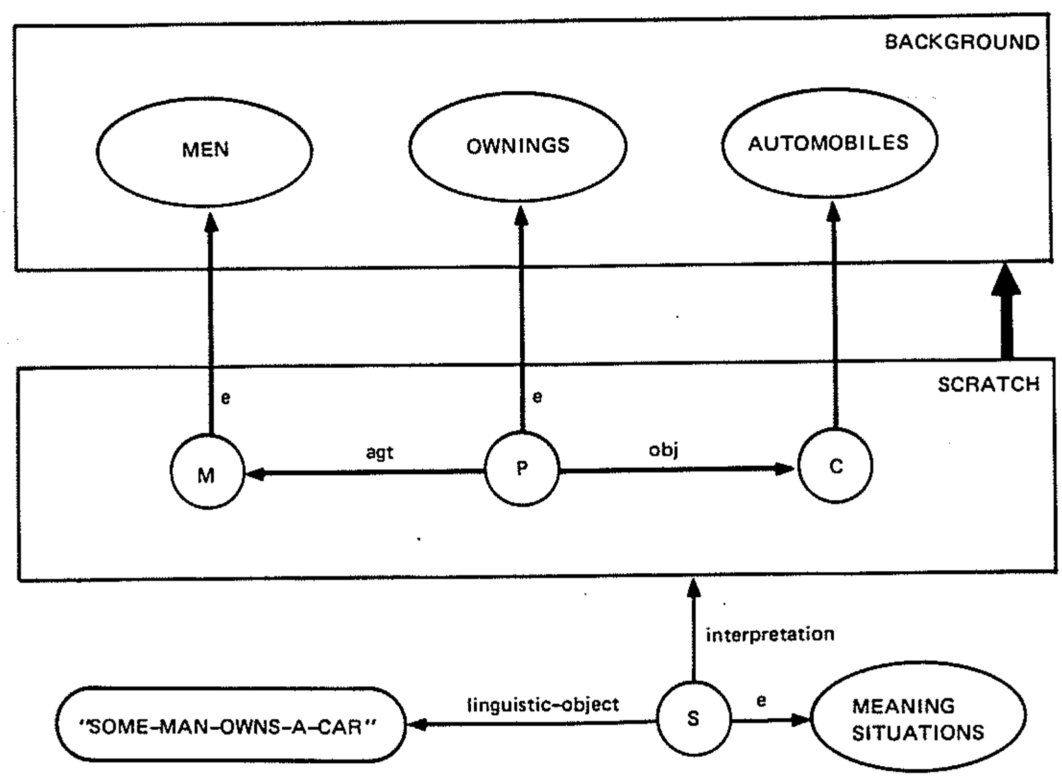


FIGURE 16 TARGET STRUCTURE FOR "SOME-MAN OWNS A-CAR"

In the translation process, spaces are created to represent the meanings of each grammatical constituent of the total utterance. These spaces are shown in Figure 6, with heavy arrows indicating the visibility hierarchy.

At the start of processing, space BACKGROUND encodes both general and some specific knowledge about Men, Automobiles, and Ownings situations. This knowledge is referenced during the translation process. For example, upon spotting the noun phrase "some-man", the natural language understanding system (NLUS) sets up a structure representing the descriptive meaning of the phrase. In particular, a new space, NP1, is created below BACKGROUND in the viewing hierarchy. Within this space, a node M is created with an e arc to node MEN in BACKGROUND. Thus, node M represents some man and the e arc makes its membership in the set Men explicit. The new space NP1 separates the structures built to represent the phrase from all other structures in the net. Similarly, new spaces V1 and NP2 are set up to encode other utterance constituents that correspond to explicit lexical entries. Note particularly that concepts conveyed by nouns and verbs are represented uniformly as elements of sets.

Once structures exist for lexical items, subphrases are grouped into larger units in accordance with the syntax. When syntax suggests combining V1 ("owns") with NP2 ("a-car") to form a larger phrase, a surface-to-deep-case map associated with the lexical entry for the verb "own" is consulted. This map indicates that an NP directly following the verb "own" generally specifies the deep obj case.*

Operating under this hypothesis, the NLUS consults the delineation of Ownings (see Section VI), which indicates that any obj of an Ownings situation must be a non-Human Physobj. The candidate for the obj is C of space NP2. Because C is an element of Automobiles, which in turn is a subset of Physobjs that is disjoint from Humans, C is accepted. (A combination such as "owns some-man" would be rejected.)

----

* The NP following "own" is not always the deep obj. Consider "What John owns John keeps".

Once V1 and NP2 have passed the acceptability test, a new space, VP1, is constructed to help encode the resultant verb phrase. This new space contains an obj arc linking node P of V1 to node C of NP2. The new arc, which constitutes the new component of meaning that VP1 adds to V1 and NP2, is visible from space VP1, but is not visible from either V1 or NP2. This leaves the components seemingly unaltered and free to combine in alternatives to VP1 if necessary.

Continuing the processing, when syntax rule R1 suggests combining NP1 with VP1 to create an S phrase (sentence), acceptability tests similar to those described above are made. When these are passed, a new space, S1, is created as shown in Figure 6, and an agt arc from P to M is placed in it. Because the new phrase is a complete sentence spanning the entire input, it is accepted as a legal interpretation.

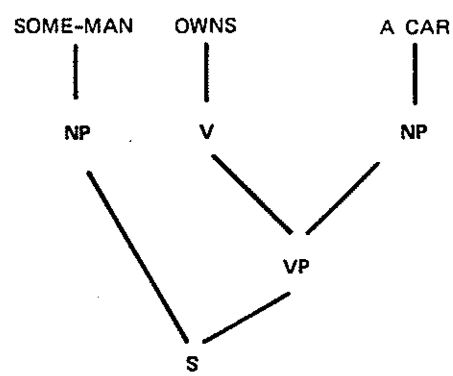


FIGURE 17 PARSE TREE FOR "SOME-MAN OWNS A-CAR

The structure built during the translation process has two particularly interesting features. First, the partial ordering of spaces from S1 to BACKGROUND is identical to that represented in Figure

17, which, because of the choice of space labels, may be recognized as the syntax tree of the sentence. Second, the view from any space shows (after subtracting out BACKGROUND) the contribution of the associated phrase to the meaning of the total input. For example, the view from NP1 shows a man M and its relationship to Men. More importantly, the view from S1 is identical to the view from SCRATCH of Figure 16.

B. Quantification

After a structure of the form described above is constructed, a second phase of the translation process copies the various nodes and arcs of the translation from the spaces that reflect the input's syntax onto structures that reflect its quantification. Because the example sentence is purely existential, nodes and arcs are copied onto the single space SCRATCH. Following conventions described in Section V, this space represents the input as a conjunction of conditions expressed solely in terms of existential variables.

However, if the input is changed from "SOME-MAN OWNS A-CAR" to

"EVERY MAN OWNS A CAR"

then the situation is different. In particular, the property list of space NP1 of Figure 6 is marked to indicate its universal quantification for the routines that copy from syntactic to quantification structures. To express the universal quantification, the copying routines create an implication structure, as described previously. Structures from the space with the universal marking are placed in the antecedent space of the implication and all structures within their scope are placed in the consequent space. The "head node" of the marked space is placed in the overlap, resulting in the structure shown in Figure 18.

Further details concerning the mechanics of this type of processing are contained in [Hendrix 1978]. Some interesting theoretical problems in translating quantifiers are discussed in [Hintikka 1973].

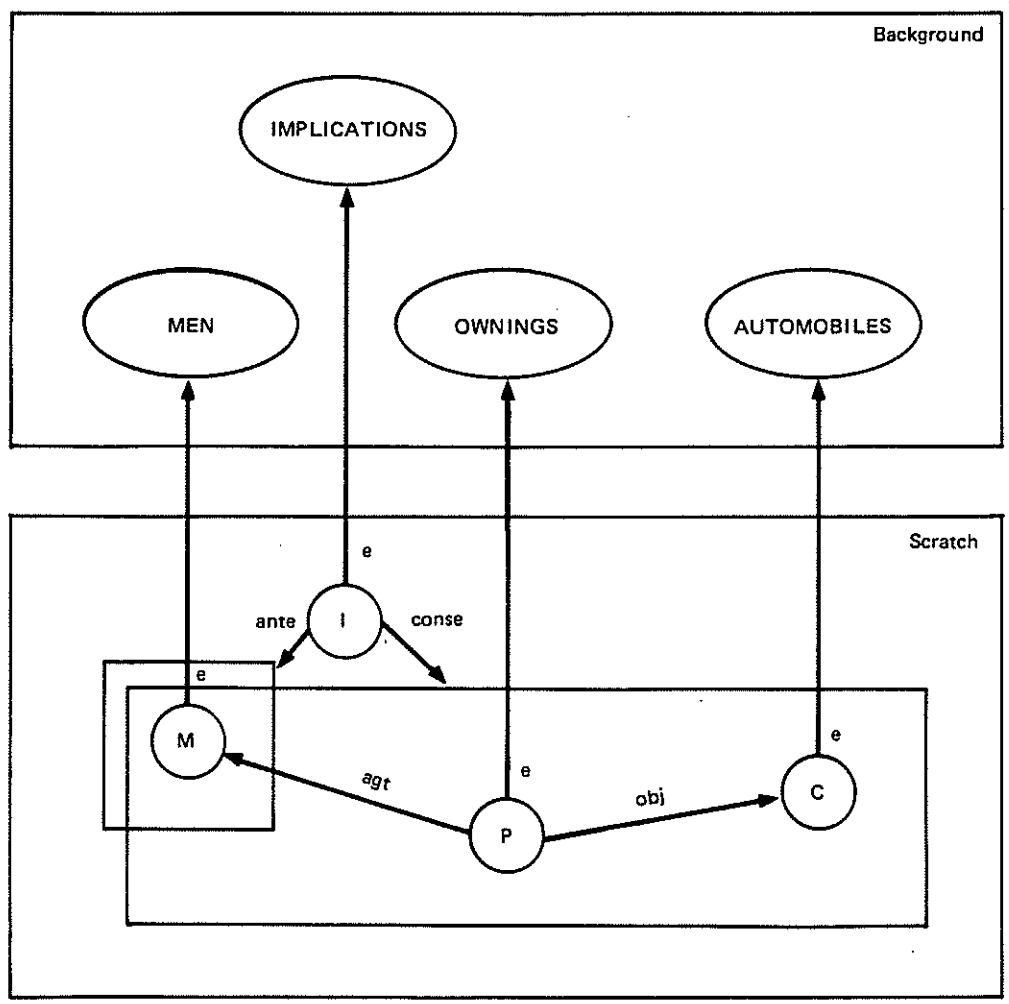


FIGURE 18 TARGET STRUCTURE FOR "EVERY MAN OWNS A CAR"

C. Resolving Definitely Determined Noun Phrases

The noun phrases in "SOME-MAN OWNS A-CAR" are indefinite in that they do not refer to any man or car in particular. However, the sentence

"JOHN OWNS THE-CAR"

contains the definitely determined noun phrases "John" and "the-car", which refer to particular objects that the hearer is expected to know about and recognize in context.

The structures built during the parsing of this sentence are shown in Figure 19. Because "John" is a proper noun whose unique referent is known to the system, reference is made directly to node JOHN in BACKGROUND. Space NP1 encompasses JOHN to indicate the interpretation of the noun phrase. Verb "own" produces space V1 as before. The definitely determined noun phrase "the-car" initially produces a structure NP2 paralleling that produced earlier for "a-car". But this time the "the" indicates that some particular car, the one currently in context, is being referenced. If Old.Black is the current topic of conversation, then Old.Black is "the-car". Hence, when Old.Black is determined to be "the-car" by means discussed below, space NP2 becomes obsolete. In its place, the NLUS constructs a new space D-NP2 around OLD.BLACK, just as it created NP1 around JOHN.

Following the syntax rules, spaces VP1 and S1 are created as before, but this time with arcs pointing directly to nodes in BACKGROUND. The view of the interpretation from S1 indicates that the new information conveyed by the sentence is that an owning situation P exists between previously known objects John and Old.Black.

The point of interest here is finding the referent of "the-car". Essentially, the problem is to find some object in BACKGROUND that both matches the description given by the phrase and is "in context". To find objects meeting the phrase description, the NLUS looks for network structures in BACKGROUND that match the network structure representing the phrase. (This task may be nontrivial, because much BACKGROUND knowledge is recorded only implicitly. For example, an e arc from

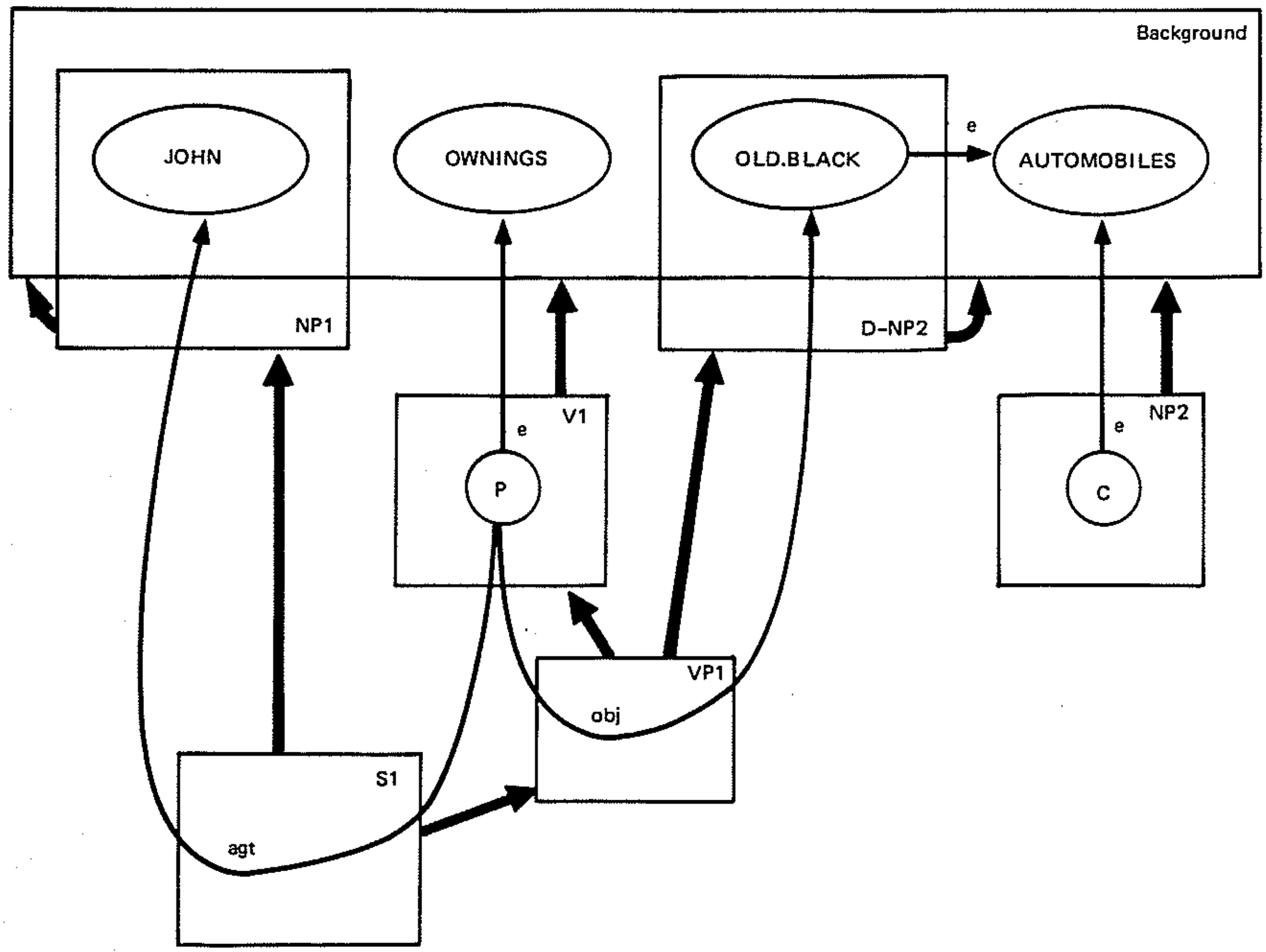


FIGURE 19 PARSING "JOHN OWNS THE CAR"

OLD.BLACK to AUTOMOBILES may be derived from an e arc from OLD.BLACK to MUSTANGS and an s arc from MUSTANGS to AUTOMOBILES. Were the problem to find "the car made by Ford", it would be necessary to derive that Ford made Old.Black from the fact that Ford made all Mustangs.) In general, there may be a great many objects in BACKGROUND meeting a given description, but only those currently in context are wanted.

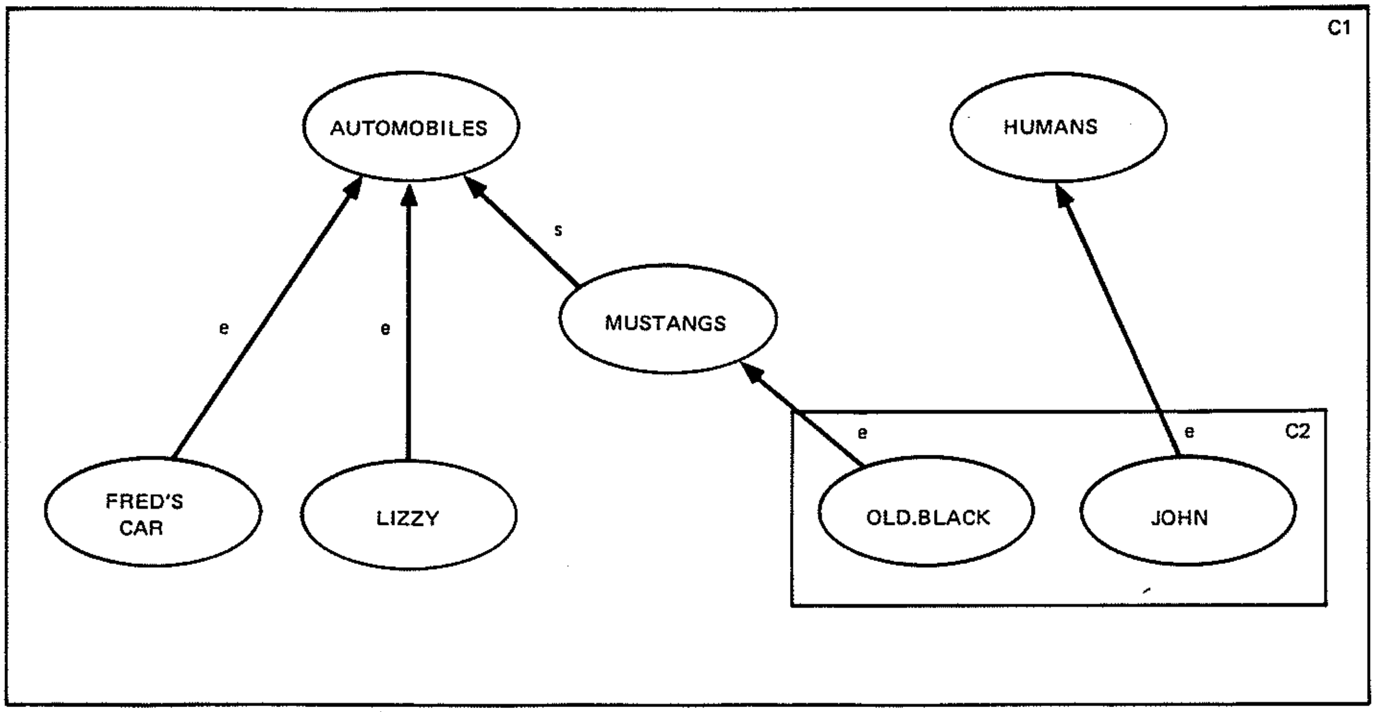


FIGURE 20 SPACES ENCODING CONTEXTS

The general problem of what objects should or should not be in the local context at any given point in an extended discourse has been investigated by a number of workers, most notably Grosz [Grosz 1977], but is beyond the scope of this paper. However, once a decision has been made concerning what belongs in a context, partitioning may be used to bundle together the objects of one context and separate them from those of another. In particular, space C2 of Figure 20 shows OLD.BLACK and JOHN grouped together into a local context that excludes such objects as FRED'S-CAR. Using vistas, a hierarchy of contexts may be defined. If the local context is encoded by space C2, and C1 is in the vista of C2, then the objects in C1 may also be considered to be in context, but at a more global level. Objects in spaces outside the vista of C2 are out of context. Following shifts in context during the course of a discourse is an important area of current research.

D. Ambiguity

The structure built during the parsing of

"SOME-MAN GAVE A-DOG A-BONE"

is shown in Figure 21. The point of interest here is that "GIVE A-DOG" is locally ambiguous in that it might mean that a dog was given to someone or that something was given to a dog. The structure of the figure reflects this ambiguity with VP1 interpreting A-DOG as filling the obj case, and VP2 interpreting A-DOG as filling the rec (recipient) case. Note that the viewing hierarchy allows both alternatives to share spaces V and NP2 without confusion, the rec arc being invisible from VP1 and the obj arc being invisible form VP2.

From the vantage of space S2, the correct interpretation of the total sentence is visible, with erroneous structures in spaces VP1 and S1 being effectively blocked out. Thus, partitioning enables networks to maintain alternative hypotheses concerning the use of input constituents and enables such competing hypotheses to share network subparts. Without partitioning or some similar technique, the back-linked nature of networks may cause a constituent to be altered when it is incorporated into a larger unit and hence render it unusable in alternative constructions.

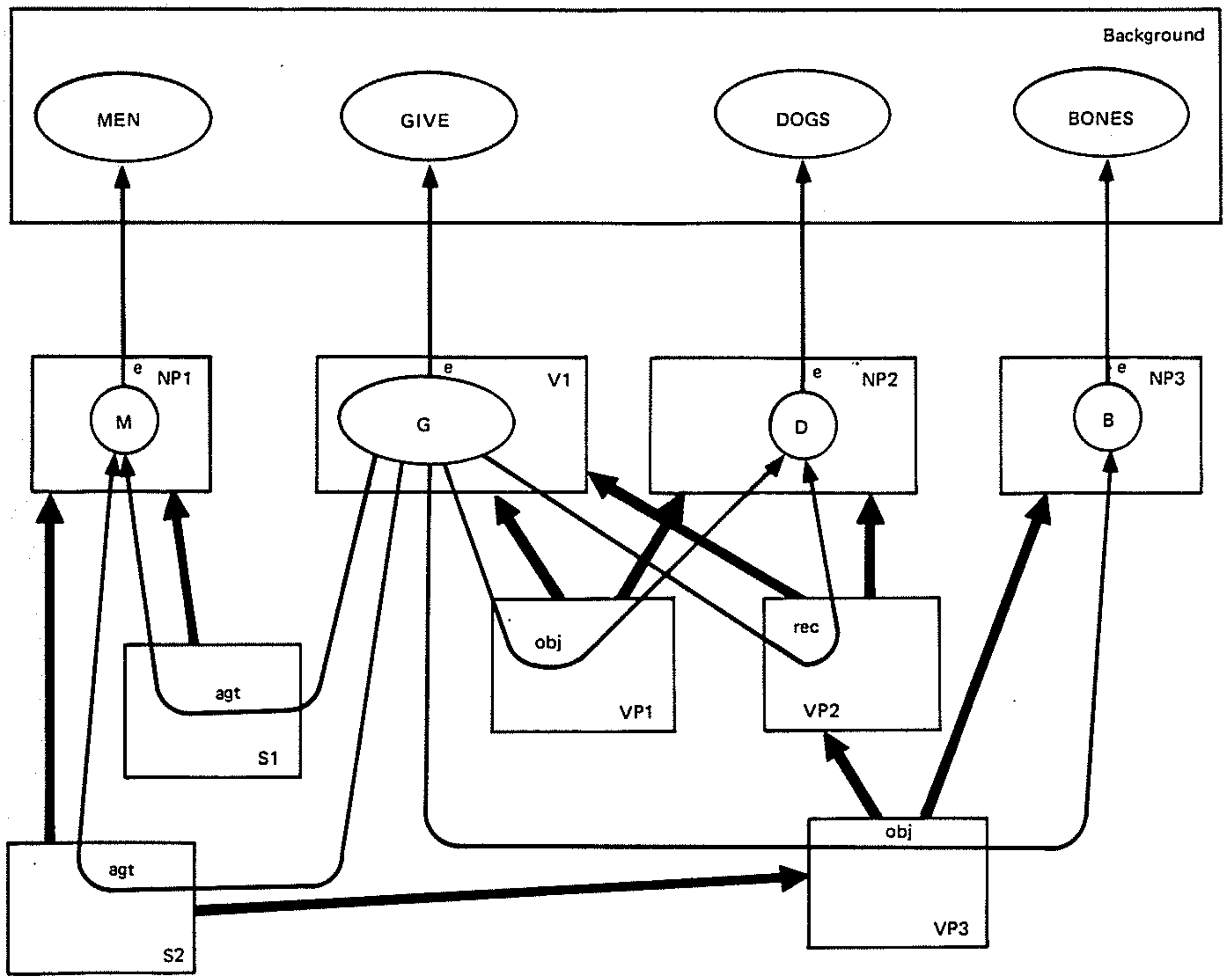


FIGURE 21 PARSING "SOME MAN GAVE A DOG A BONE"

X LINEARIZED NET NOTATION

To provide a convenient formalism for communicating network structures to the computer, a linearized net notation, called the <u>LN2</u> language, has been devised as an extension of INTERLISP [Teitelman 1975]. The syntax of LN2 was inspired by and bears some resemblance to the syntax of KRL [Bobrow and Winograd 1977].

The flavor of this language is indicated by the following statement, which builds the network of Figure 22.

```
(!SPACE S1
     [UNIVERSAL]
     [SITUATIONS (ARE UNIVERSAL)]
     [IMPLICATIONS (ARE SITUATIONS)]
     [OWNINGS (ARE SITUATIONS)]
     [SUBMARINES (ARE UNIVERSAL)]
     [LAFAYETTES (ARE SUBMARINES)]
     [HENRY.L.STIMSON (A LAFAYETTE)]
     [COUNTRIES (ARE UNIVERSAL)
                (SINGULAR COUNTRY)]
     [THE.U.S. (A COUNTRY)]
     [x (AN OWNING)
        {agt THE.U.S.}
        {obj HENRY.L.STIMSON}]
     (TURN.OFF.D)
     (IMPLICATION
         ([U (A SUBMARINE)])
         ([(AN OWNING)
           {obj U}
           {agt (A COUNTRY)}]))
     (LN2.SUB
           owns (OWNER OWNEE)
             [(AN OWNING)
               {agt OWNER}
               {obj OWNEE}])
     (IMPLICATION
         ([y (A LAFAYETTE)])
         (<owns THE.U.S. y>))  )
```

The total statement is a call to function !SPACE of the form (!SPACE name e1 e2 ... en). Its first argument is a name to be given to a newly created space. All subsequent arguments are expressions to be executed in the context of the new space.

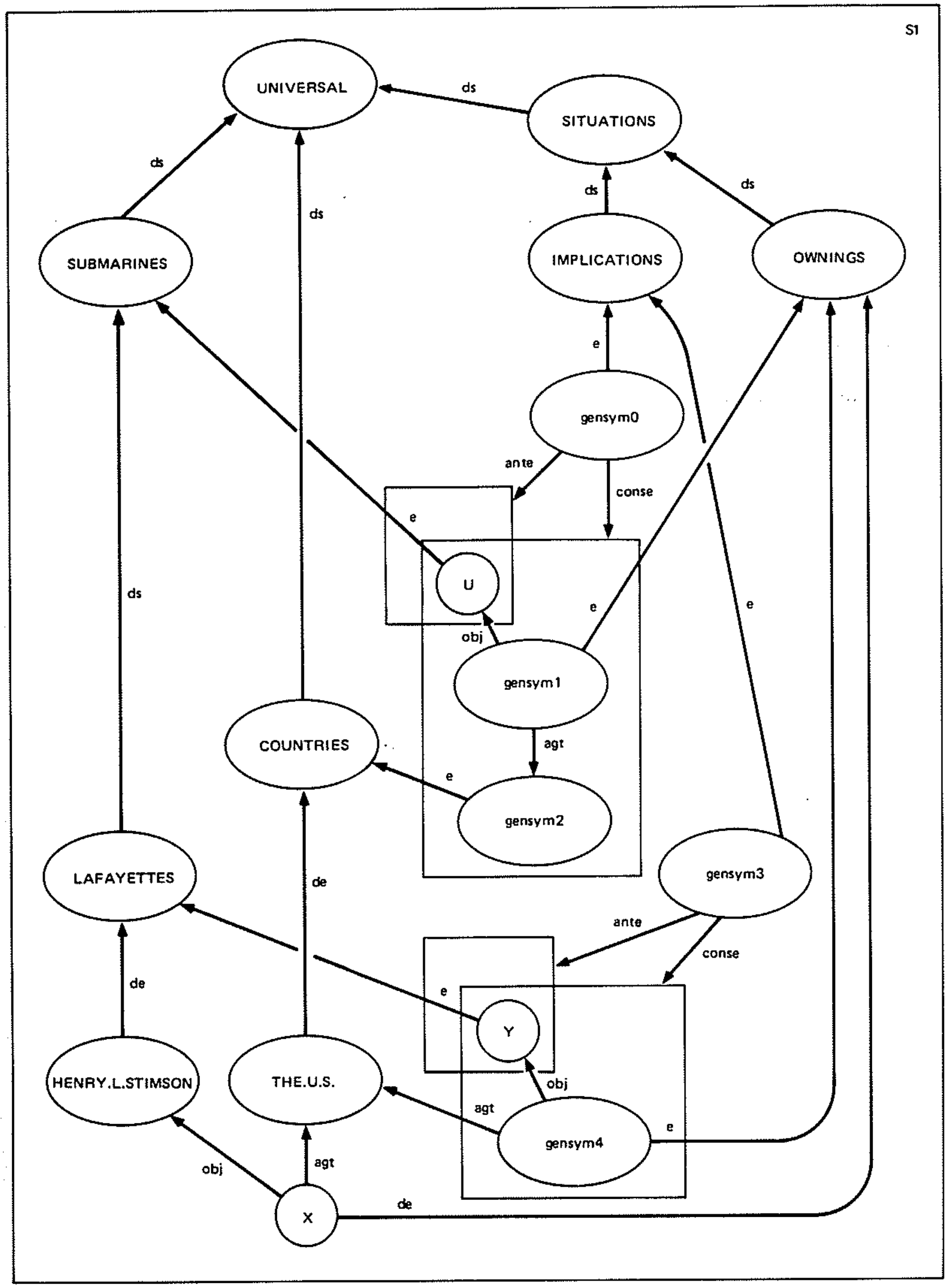


FIGURE 22 NETWORK CREATED BY LN2

The first such expression is "[UNIVERSAL]", which read-macros expand into "(!NODE UNIVERSAL)". In general, calls to !NODE are of the form (!NODE optional-name e1 e2 ... en). The function creates a new node on the current space, assigns it the optional-name, and then evaluates the various expressions ei. Thus, [UNIVERSAL] just creates a node named UNIVERSAL.

"[SITUATIONS (ARE UNIVERSAL)]" creates a node named SITUATIONS and then executes the expression "(ARE UNIVERSAL)", which creates a ds arc from the current node to UNIVERSAL. The next four !NODE expressions are similar.

"[HENRY.L.STIMSON (A LAFAYETTE)]" causes a node to be created named HENRY.L.STIMSON. Function A produces a de arc from this node to LAFAYETTES, the node whose name is formed by adding "S" or "ES" to the argument of A.

Since COUNTRY has an irregular plural, the expression creating node COUNTRIES has a call to function SINGULAR to note this fact. SINGULAR does the necessary bookkeeping so that the call to A in "[THE.U.S. (A COUNTRY)]" works properly.

"[x (AN OWNING) {agt THE.U.S.} {obj HENRY.L.STIMSON}]" creates node x, encodes x as a distinct element of OWNINGS, and then creates an agt arc to THE.U.S. and an obj arc to HENRY.L.STIMSON.*

The expression "(TURN.OFF.D)" changes the operation of functions A and ARE so that de and ds arcs are replaced subsequently by e and s arcs.

Function IMPLICATION takes two arguments: a list of expressions for creating structures inside an implication ante space and a similar list for the conse space. IMPLICATION builds a new element of IMPLICATIONS with appropriate new spaces and then executes the lists of expressions. New structures created or referred to by both ante and conse are placed in the overlap.

----

* A delineation for the set of Ownings situations would be included in a more complete network, but is omitted here.

In the first IMPLICATION of the example, the ante space expressions (there is only one) cause a node labeled "U" to be created with an e arc to SUBMARINES. The sole conse space expression calls for a node to be created and assigned a gensym name. The node represents an element of OWNINGS. The obj of this element is U. The agt is to be encoded by a newly created, gensym-named node with an e arc to COUNTRIES.

Function LN2.SUB creates no structure itself but defines an LN2 subroutine for subsequent use. The example shown defines a subroutine called "owns" that may be used to create instances of Ownings situations in terms of the local variables (formal parameters) OWNER and OWNEE. The remaining arguments to LN2.SUB are expressions to be evaluated when the "owns" subroutine is invoked. An invocation of the "owns" subroutine occurs in the conse of the last IMPLICATION. The delimiters "<" and ">" indicate that an LN2 subroutine is to be invoked. The first argument within the delimiters is the subroutine name (which must have been previously declared in a call to LN2.SUB) and the other arguments are actual parameters. Calls to LN2 subroutines are designed to resemble expressions of propositions in predicate calculus notation.

XI IMPLEMENTATION

SRI's implementation of partitioned networks is written in INTERLISP [Teitelman 1975] and makes extensive use of user data types. Nodes, arcs, and spaces are each represented by a separate record type.

A space record is a collection of four pointers referencing the list of nodes in the space, the list of arcs, the property list, and the node structure that encodes the space's node-like properties if it is a supernode. The latter field is NIL for spaces that are not supernodes.

A node record is a collection of six pointers referencing the incoming arcs, the outgoing arcs, the node label (if assigned), the spaces upon which the node lies, the property list, and the associated space if the node is part of a supernode complex.

An arc record is a collection of five pointers referencing the arc label, the "from" node, the "to" node, the spaces upon which the arc lies, and the arc's property list.

The most recent versions of the net package, including one version that maintains large nets on secondary storage, were programmed by Jonathan Slocum, Ann Robinson, and Kurt Konolige. The largest partitioned net created to date contains some 4500 network structures (nodes, arcs, and spaces) and was used in the PROSPECTOR rule-based inference system. Models using in the neighborhood of 10000 structures are under construction.

XII CONCLUSION

This paper has outlined the basic concepts underlying the encoding of knowledge in partitioned networks and has discussed structures used for a variety of specialized applications. The ability of networks to encode knowledge in a form convenient for a diversity of applications enhances the value of networks as a medium for integrating many skills in one coordinated system. Partitioning has increased the usability of networks for those applications in which a subportion of a network is to be treated collectively as a unit.

XIII ACKNOWLEDGEMENTS

Work on partitioned networks and their applications has been supported in part by the National Science Foundation under Grant No. MCS76-22004, the Defense Advanced Research Projects Agency under Contracts DAHC04-75-C-0005, DAAG29-76-C-0011, DAAG29-76-C-0012 and N00039-78-C-0060, and the Office of Resources of the U. S. Geological Survey under Contract No. 14-08-0001-15985. Thoughtful critiques of early drafts of this paper were given by Robert Moore. Virtually everyone in the Natural Language, Prospector and Decision Aids projects

at SRI International has made contributions to the development of partitioned networks. It is hoped that feedback from this community of users will continue to support the evolution of more powerful systems.

REFERENCES

Bobrow, D. G., and Winograd, T. 1977. "An Overview of KRL, A Knowledge Representation Language." Cognitive Science. 1, 1977, pp. 3-46.

Brachman, Ronald J. 1977. "What's in a Concept: Structural Foundations for Semantic Networks." International Journal of Man-Machine Studies. 9, March, 1977, pp. 127-152.

Brachman, Ronald J. 1978. "On the Epistemological Status of Semantic Networks." In Associative Networks - The Representation and Use of Knowledge in Computers. Nicholas V. Findler (ed.), forthcoming, 1978.

Bruce, Bertram C. 1973. "A Model of Temporal References and Its Application in a Question Answering Program." Artificial Intelligence. 3, 1973, pp. 1-26.

Cohen, Philip R., and Perrault, C. R. 1976. "Preliminaries for a Computer Model of Conversation." In Proceedings of the First CSCSI/SCEIO National Conference, Vancouver, British Columbia, Canada, August 25-27, 1976, pp. 102-111.

Duda, R. O., Hart, P. E., Nilsson, N. J., and Sutherland, G. L. 1977. "Semantic Network Representations in Rule-Based Inference Systems," Technical Note 136, Artificial Intelligence Center, SRI International, Menlo Park, California, March 1977. To appear in Pattern Directed Inference Systems, D. A. Waterman and F. Hayes-Roth (eds.), Academic Press, 1978.

Duda, R. O., Hart, P. E., and Reboh, R. 1977. "A Rule-Based Consultation Program for Mineral Exploration." Proceedings of the Lawrence Symposium on Systems and Decision Sciences, pp. 306-309, Berkeley, California, October 2-4, 1977.

Fahlman, S. E. 1977 "A System for Representing and Using Real-World Knowledge." Ph.D. Dissertation, MIT, September, 1977.

Fikes, R. E., Hart, P. E., and Nilsson, N. J. 1972. "Learning and Executing Generalized Robot Plans." Artificial Intelligence 3, 251-288.

Fikes, R. E., and Hendrix, G. G. 1977. "A Network-Based Knowledge Representation and its Natural Deduction System." Fifth International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, August 22-25, 1977, pp. 235-246.

Fillmore, Charles J. 1968. "The Case for Case." In Universals in Linguistic Theory. E. Bach and R. Harms (eds.), 1968, pp. 1-88.

Grosz, B. J. 1977. "The Representation and Use of Focus in Dialog Understanding," Ph.D. Dissertation, U. C. Berkeley, May, 1977.

Hayes, Philip J. 1977. "Some Association-Based Techniques for Lexical Disambiguation by Machine." Doctoral Thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 1977.

Hendrix, Gary G. 1973. "Modeling Simultaneous Actions and Continuous Processes." Artificial Intelligence. 4, 1973, pp.145-180.

Hendrix, Gary G. 1975a. "Expanding the Utility of Semantic Networks Through Partitioning." Advance Papers, International Joint Conference on Artificial Intelligence, Tbilisi, Georgian SSR, September 3-8, 1975, pp. 115-121.

Hendrix, Gary G. 1975b. "Partitioned Networks for the Mathematical Modeling of Natural Language Semantics." Technical Report NL-28, Department of Computer Sciences, University of Texas, Austin, Texas, 1975.

Hendrix, Gary G. 1978. "Semantic Aspects of Translation." In Understanding Spoken Language. D. E. Walker (ed.), Elsevier, New York, 1978, pp. 193-226.

Hintikka, Jaakko. 1973. "Quantifiers vs. Quantification Theory." Linguistic Inquiry. 5, 1974, pp. 153-177.

Kay, Martin. 1973. "The Mind System." In Natural Language Processing. Randall Rustin (ed.), Algorithmics Press, New York, 1973, pp. 153-188.

McCarthy, J. 1968. "Situation, Actions, and Causal Laws." In Semantic Information Processing. Marvin Minsky (ed.), MIT Press, Cambridge, MA., 1968, pp. 410-417.

Montague, R. 1974. "Syntactical Treatments of Modality, with Corollaries on Reflexion Principles and Finite Axiomatizability." In Formal Philosophy. R. H. Thomason (ed.), Yale University Press, New Haven, 1974, pp. 286-302.

Moore, Robert C. 1975. "Reasoning from Incomplete Knowledge in a Procedural Deduction System." Technical Report 347, Artificial Intelligence Laboratory, MIT, Cambridge, MA., December, 1975.

Norman, Donald A., Rumelhart, David E., and the LNR Research Group. 1975. Explorations in Cognition. Freeman, San Francisco, 1975.

Reboh, R., and Sacerdoti, Earl D. 1973. "A Preliminary QLISP Manual." Technical Note 81, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, August, 1973.

Sacerdoti, Earl D. 1977. A Structure for Plans and Behavior. Elsevier North-Holland, New York, 1977.

Schubert, L. K. 1976. "Extending the Expressive Power of Semantic Networks." Artificial Intelligence. 7, 1976, pp. 163-198.

Scragg, Greg W. 1975a. "Answering Questions About Processes." In Explorations in Cognition. Donald A. Norman and David E. Rumelhart (eds.), W.H. Freeman, San Francisco, 1975, pp. 349-375.

Scragg, Greg W. 1975b. "Frames, Planes and Nets: A Synthesis." Technical Note 19, Istituto per gli Studi Semantici e Cognitivi, Castagnola, Switzerland, 1975.

Shapiro, Stuart C. 1971. "A Net Structure for Semantic Information Storage, Deduction and Retrieval." Advance Papers, Second International Joint Conference on Artificial Intelligence, London, England, September 1-2, 1971. The British Computer Society, London, England, 1971, pp. 512-523.

Shortliffe, E. H. 1976. Computer-Based Medical Consultations: MYCIN. Elsevier, New York, 1976.

Simmons, Robert F. 1973. "Semantic Networks: Their Computation and Use for Understanding English Sentences." In Computer Models of Thought and Language. Roger C. Schank and Kenneth M. Colby (eds.), Freeman, San Francisco, 1973, pp. 63-113.

Teitelman, Warren. 1975. "INTERLISP Reference Manual." XEROX Palo Alto Research Center, Palo Alto, California, 1975.

Walker, Donald E. (Ed.) 1978. Understanding Spoken Language. Elsevier North-Holland, New York, forthcoming, 1978.

Woods, William A. 1970. "Transition Network Grammars for Natural Language Analysis." CACM, 13, 1970, pp. 591-606.

Woods, William A. 1975. "What's in a Link: Foundations for Semantic Networks." In Representation and Understanding. Daniel G. Bobrow and Allen M. Collins (eds.), Academic Press, New York, 1975, pp. 35-82.