American Journal of Computational Linguistics *Microfiche 34*

PROCEEDINGS

13TH ANNUAL MEETING

ASSOCIATION FOR COMPUTATIONAL LINGUISTICS

3 : PARSING, SYNTAX, AND SEMANTICS

Timothy C. Diller, Editor

Sperry-Univac
St. Paul, Minnesota 55101

PREFACE

The talks presented in Session 3 of the 13th ACL meeting dealt with three facets of the language understanding process: the integration of knowledge sources (Paxton and A. Robinson), syntactic processing (Bates and J Robinson), and semantic processing (Hendrix, Sondheimer and Perry, and Cercone). The first four talks were oriented toward speech input while the talk by Sondheimer and Perry assumed phrase structure trees as input and the talk by Cercone assumed textual input. The paper from which Bates's talk was drawn was much too extensive for inclusion here and will be published separately My special thanks to Stan Petrick who agreed to chair this session on very short notice.

--Timothy C. Diller
Program Committee Chairman

TABLE OF CONTENTS

SESSION 3: PARSING, SYNTAX, AND SEMANTICS

SYNTACTIC PROCESSING IN THE BBN SPEECH UNDERSTANDING SYSTEM

MADELINE BATES
Bolt Beranek and Newman Inc.

The syntactic analysis system presented here is composed of two parts, a modified augmented transition network grammar and a parser which is designed for a speech understanding environment.

The parser operates on partial utterances called theories. A theory may be thought of as a set of words which are hypothesized to be in the utterance. The parser processes the words in a theory by building partial syntactic paths using the words of the theory. These paths do not depend on left context, which will be missing if there are gaps in the theory. Syntactic constituents are built where possible and, whenever a constituent is built, the parser can interface with the semantic component of the total speech understanding system for guidance and verification.

The parser tries to predict words and/or syntactic categories to fill or reduce gaps in the theory, particularly small function words which are difficult to detect reliably on acoustic grounds alone. The parser does not follow all possible parse paths, but attempts to select the most likely ones for extension. It uses a judicious mixture of top down, bottom up, depth first, and breadth first parsing strategies to take advantage of local, reliable information. It saves all the information gained while following alternative parse paths, so that several parse paths which share a common part, even if the paths are in different theories, can share that portion without reparsing. This is true even if the parse paths split before and/or after the common part and even if the common section analyzes only part of a syntactic constituent.

American Journal of Computational Linguistics *Microfiche 34 : 5*

SYSTEM INTEGRATION AND CONTROL IN A SPEECH UNDERSTANDING SYSTEM

WILLIAM H. PAXTON AND ANN E ROBINSON

Artificial Intelligence Center
Stanford Research Institute
Menlo Park, California 94025

ABSTRACT

Two important problems in speech understanding are how to effectively integrate multiple sources of knowledge within the system and how to control the activities of the system to arrive at appropriate interpretations for utterances. This paper first describes the roles played by acoustics, syntax, semantics, and discourse, and shows how a language definition is used to integrate them into a system in a way that allows the interactions to be easily visible. The second part of the paper describes an executive that uses information from these knowledge sources in its control strategy.

Acknowledgment

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and monitored by the U.S. Army Research Office under Contract No. DAHC04-75-C- 006.

A speech understanding system must use many kinds of knowledge, each playing a particular role during the interpretation of an utterance. While these roles are interrelated, it is important to be able to separate the knowledge sources so that interrelations are visible and so that the contributions from the various sources can be studied. The knowledge sources used in the system being developed jointly by SRI and SDC can be characterized broadly under the headings of acoustics, syntax, semantics, and discourse (Walker et al., 1975; Robinson, 1975; Hendrix, 1975; Deutsch, 1975; Slocum, 1975; Pitea, 1975).

The acoustic component relates linguistic entities (words and phrases) to the speech waveform. An acoustic-phonetic processor analyzes the digitized waveform to extract parameters based on speech production characteristics. The parameters include fundamental frequency, voicing label, formant frequency, energy data, and others. Following parameterization, various rules are applied to generate an acoustic feature description of the utterance. The parameters and features are subsequently used by the lexical mapping procedure. The mapper is called during the parsing of an utterance to give a decision score as to whether a proposed word or phrase could actually be present in a specified time region of the input. Phonological and acoustic-phonetic rules are used by the mapper to relate phonetic spellings to acoustic data.

Syntax provides reliable, reasonably inexpensive indications of which words or groups of words may combine and of how well they fit. Syntactic rules give general patterns for constructing noun phrases, clauses, and sentences and provide consistency checks for such items as number agreement. In testing word or phrase combinations, syntactic information alone can often rule out a candidate without the need for more costly semantic and discourse analysis.

The semantic component includes a general model of the domain of discourse, and a set of algorithms for combining (or rejecting) concepts in the domain. For example, given a verb and two noun phrases, semantic routines can build the corresponding semantic relation between the items indicated by the noun phrases.

The discourse component deals with the relationship of the current utterance (or a portion of it) to the dialog context and to entities in the task domain. Discourse functions use information from previous utterances to fill out elliptical expressions and to find referents for pronouns and definite noun phrases.

The language definition is the focal point for integrating these knowledge sources. A language definition includes (1) sets of units out of which utterances in the language are constructed and (2) rules for combining the units into larger structures. The basic units will be called 'words' (although this technical use does not exactly correspond to the common use). The composition rules indicate how phrases can be combined into still larger phrases. More precisely, a 'phrase' is either a word in the input or the result of applying a composition rule to constituent phrases. The rules give the linear pattern of constituents and specifications for calculating values for both the attributes of the resulting phrase and for factors used in judging the result.

It is at the phrase level that the knowledge sources are integrated into the system. There are two aspects to the contributions from each source: the values of properties of the phrase as computed by the knowledge source, and the source's assessment of the correctness of this phrase as an interpretation of the input. These two aspects are reflected in the attribute and factor statements that are associated with each of the words and phrases in the language definition. The attribute statements provide instructions for computing various properties of the phrase. These instructions may call upon any or all of the sources of knowledge. For example for a phrase spanning a particular segment, an acoustic attribute may specify the words in that segment; an attribute supplied by the syntax can specify a feature such as the voice ('active' or 'passive'); an attribute supplied by semantics can specify a semantic net interpretation built from the semantics of the constituents; and an attribute supplied by the discourse component can indicate a referent or an implied meaning.

Factor statements tell how to use these attributes in determining the likelihood that the phrase is a correct interpretation of the input. The result of combining the factors for a particular phrase is called a score. The use of such scores by the executive in determining overall strategy is described below. Factors are nonbinary; since they can have a range of values, rigid 'yes' or 'no' decisions do not have to be made in assessing the quality of a phrase. For example, the closeness of the acoustic match may vary and this can be reflected in the corresponding factor. Weak evidence from one source of knowledge could lower the score, while strong evidence from another source could compensate for that and actually raise the score.

In summary, a phrase is a composite interpretation of a particular portion of the utterance, integrating contributions from all relevant knowledge sources. This means that each portion of the input is interpreted and evaluated by the system as fully as possible, as soon as possible. The system is never faced with the problem of relating or combining fragmentary theories constructed independently by different knowledge sources, and evaluations made by different sources are immediately merged to control and coordinate overall system activity. For example, as soon as a definite noun phrase is found, the acoustic component checks the coarticulation of the constituents, the syntactic component checks for agreement in features such as number, the semantic component builds a representation of the meaning, and the discourse component looks for a referent.

The following example illustrates how several knowledge sources are used together to interpret and evaluate phrases. The rule shown is for the composition of a noun phrase such as 'what submarine' or 'their submarines' and illustrates the integration of acoustic, syntactic, semantic, and discourse information.

```
RULE.DEF NP7 NP = DET NOM;

  ATTRIBUTES

    STRING = APPEND(STRING(DET),STRING(NOM)),
    NBR = GINTERSECT(NBR(DET),NBR(NOM)),
    CMU = GINTERSECT(CMU(DET),CMU(NOM)),
    SEMANTICS = SEMCALL("SEMRNP7,SEMANTICS(NOM),
      MOOD(DET),GCASE(DET),INTERPRETATION(DET)),
    DISCOURSE = IF MOOD(DET) EQ "DEC THEN
      DISCALL("DISRNP7,SEMANTICS) ELSE "UNDEFINED,
    INTERPRETATION = IF DISCOURSE NQ "UNDEFINED THEN
      DISCOURSE ELSE SEMANTICS;

  FACTORS

    COART = MAPPER(LASTWORD(DET),FIRSTWORD(NOM)),
    NBR = IF NULL(NBR) THEN OUT ELSE OK,
    CMU = IF NULL(CMU) THEN OUT ELSE OK,
    SEMANTICS = IF NULL(SEMANTICS) THEN OUT ELSE OK,
    DISCOURSE = IF MOOD(DET) NQ "DEC THEN OK ELSE
      IF NULL(DISCOURSE) THEN POOR ELSE
      IF AMBIGUOUS(DISCOURSE) THEN OK ELSE GOOD;

END;
```

The first attribute statement computes the STRING of the resultant phrase, which is an acoustic attribute indicating the words composing this phrase. NBR (number) and CMU (count-mass-unit) are syntactic attributes for the phrase, each being derived from the intersection of the corresponding attributes of the constituents. The semantics attribute is a piece of semantic net that is constructed from the semantics of the constituents by the semantic routine (SEMRNP7) associated with this rule. If the MOOD attribute of the DET constituent is "DEC, i.e., a declarative determiner, then the discourse routines will look for a referent for the phrase in the dialog context and assign its semantic structure as the value of the attribute DISCOURSE. The INTERPRETATION of the phrase is either the referent found by discourse or the semantic net structure in case no direct reference is found.

The factor statements use these attributes in computing contributions towards the score for the phrase. As has been mentioned, there is a range of acceptable values for factors. For simplicity, symbolic values are used (VERYGOOD, GOOD, OK, POOR, BAD, and OUT). In the example rule, there are factors determined by each of the major knowledge sources. The COART factor reflects an acoustic test of the coarticulation of the last word of the determiner and the first word of the nominal. NBR and CMU are syntactic factors that will eliminate the phrase if either attribute is incompatible between the constituents. The semantic factor will eliminate the phrase if no semantic interpretation can be formulated. While the current semantic component does not have a metric for determining the likelihood of an interpretation other than whether or not a semantic representation can be built, it is possible to introduce such a metric and have the semantic factors be nonbinary. The discourse factor is nonbinary. If the determiner is declarative, the discourse has tried to find a referent. If no referent was found, the factor is given a low value, 'POOR', but the phrase is not discarded. If several possible referents were found, the phrase is kept and the score is not lowered because the ambiguity can perhaps be resolved later. If just one referent was found, it is taken as evidence that the phrase is a correct interpretation for that portion of the utterance and the factor is given a higher value 'GOOD'.

The example discussed above shows how the language definition system can be used to integrate a variety of knowledge sources in a way that keeps the contributions and interactions of the different sources easily visible. The representation combines procedural information (in the expressions for calculating attribute and factor values) and declarative information (in the constituent pattern) in a form designed to simplify the task of writing a large definition containing many rules. However, before the rules can actually be used, they must be converted to a different representation designed with efficiency in mind. This translation is done by a language definition compiler' that constructs an internal representation of the language definition that depends in an intricate way on the structure of the 'executive', the portion of the system responsible for scheduling and controlling the various tasks to be performed in constructing an interpretation of an utterance. The operation of the executive is the subject of the rest of this paper.

The executive makes a distinction between the phrases being built and the tasks required to build these phrases. A data structure, called the 'parse net', represents the growing collection of phrases, and another structure, called the 'task queue', encodes the alternative operations available for taking another step toward understanding the input. Each entry in the task queue specifies a procedure to be performed at a particular location (node) in the parse net. The performance of such a procedure typically entails both modifying the parse net and scheduling new tasks to make further modifications. Each task has associated with it a priority for performing it. The method for determining priorities is described below.

Tasks can include looking for a new word or phrase to finish an incomplete phrase (one missing some of its constituents) and trying to use a word or completed phrase in a larger phrase. This means that the system can work both 'top down' and 'bottom up', because it can look in a goal-driven manner for missing constituents of higher level phrases, and it also can accept words from the acoustics to build into larger phrases in a data-driven manner. As an example, consider the simple grammar with the following composition patterns:

S = NP VP
VP = VP NP | VERB
VERB = own | lost
NP = they | the house | it

Assume that the word 'they' has been found initially either by the acoustics directly or as a result of confirming a prediction made by the language definition. 'They' constitutes a complete NP. This NP can be put into the S rule, causing the partially filled phrase 'they VP' to be added to the parse net. Already, some of the attributes and factors for the S rule can be determined, and a score computed for this phrase. Building this partial phrase leads to the creation of a new task: to look for a VP following the NP. That task in turn leads to two alternative subtasks: look for a VP NP or look for a VERB. Priorities for both these tasks are computed and they are put on the task queue to be processed. The executive then removes the next task from the queue and continues.

In general, deciding which task to perform is of great importance, because only a subset of the scheduled tasks will actually prove to be necessary to understand the input; the others will be 'false steps' leading to dead ends. Ideally, in deciding which task to do, the executive would always choose one of the necessary tasks and never take a false step. The utterance would be understood with the unnecessary tasks still left in the queue. To approach this ideal, the actual system must spend some of its effort in choosing tasks. Such effort is well spent if it produces a net decrease in processing time. In other words, the efficiency of the system will be improved by decisions regarding the order in which tasks are performed, if the cost of the decisions is less than the cost of the false steps that would otherwise have been taken. Since acoustic uncertainty in speech understanding makes the potential for wasting effort on unnecessary operations particularly large, the system can afford to carry out rather complex computations in deciding what to do next and still obtain a large improvement in overall efficiency. In the current system, the decisions are based on the relative priorities assigned to the various tasks waiting in the queue. Tasks are associated with phrases, and task priorities largely depend on how important the system feels it is to process the phrase.

In addition to the scores of phrases, which combine a variety of factors but are independent of the larger sentential context, the system forms another assessment of the quality of the phrase called the phrase 'value', which depends on the context of proposed complete interpretations for the entire utterance. The phrase value is an estimate of the highest score for all possible interpretations spanning the utterance that include the phrase. The estimate is computed by means of a heuristic search of the space of possible sentential contexts established during the previous tasks performed by the executive.

The priority of a task is initially set to the value of its associated phrase, but the priority is lowered if the task conflicts with the executive's current 'focus of activity'. The phrase value that determines the initial priority reflects an evaluation of both the internal structure of the phrase and its relation to its context, but it does not reflect its competition. If a phrase has a high value, other similar phrases are also likely to have high values. If values alone determined priorities, then even after successfully completing a phrase, the system would tend to continue looking for minor variations in the same area rather than moving on to look for ways to construct a complete interpretation. The 'focus of activity' mechanism provides a way for phrases to inhibit the executive from looking for competing phrases that would necessarily replace them. This focusing is brought about by lowering the priority of tasks that look for replacements for any of a set of focus phrases, until the potential replacement promises to lead to a significant improvement in value for the final interpretation. The effect is to bias the executive toward building up a complete interpretation using phrases in focus rather than exploring competing interpretations that would not use focus phrases. If the focus is wrong, the attempts to extend it to a complete interpretation will be unsuccessful. Eventually a task that conflicts with the focus will become the highest priority operation for the executive to perform in spite of the bias against it. As a result the focus will be modified so that it is consistent with the new task, and the executive will then concentrate on using the revised set of phrases.

In addition to calculating priorities of tasks on the basis of phrase values and focus of activity, the executive must ensure that the information gained through the performance of the tasks is used effectively. This is done by structuring the parse net and the tasks that operate on it in a way that brings together related activities and coordinates them to eliminate duplication of effort. By avoiding duplication, the system reduces the ill effects of the false steps it will inevitably take. Work done on a false path is not necessarily wasted, since it may produce a phrase that can be used in some other way. For example, a phrase constructed as part of an unsuccessful search for one type of sentence may later appear in the final interpretation as part of a different kind of sentence. Also, false steps are not repeated, since the system only makes one attempt to build a particular type of phrase in a particular location in the utterance, regardless of how many larger phrases might include it. Mistakes are inevitable, but at least the system will not make the same mistake twice in one parse

To summarize, the language definition is designed to facilitate the integration of many knowledge sources. Rules in the language definition contain attributes and factors from all of these sources. The attributes are used to indicate particular properties of phrases, and factors then use these attributes to determine the score of the phrase. The external representation of the language, designed for easy use by people, is converted by a language definition compiler into an internal representation, designed for efficient use by the executive. In a step by step manner, the executive uses this information to create, evaluate, and combine phrases. The choice of the next operation to carry out takes the form of assigning priorities to alternative tasks. Priorities reflect both the expected values of complete interpretations toward which the task would lead and the relation of the task to the current focus of activity. Finally, the entire process is organized so that information gained in performing a task is shared and recorded in such a way that it does not have to be rediscovered.

References

Deutsch, Barbara G. Establishing Context in Task-Oriented Dialogs. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Hendrix, Gary G. Semantic Processing for Speech Understanding. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Ritea, H. Barry. Automatic Speech Understanding Systems. Proceedings, Eleventh Annual IEEE Computer Society Conference, Washington, D.C., 9-11 September 1975.

Robinson, Jane J. A Tuneable Performance Grammar. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Slocum, Jonathan. Speech Generation from Semantic Nets. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Walker, Donald E., et al. Speech Understanding Research. Annual Report, Project 3804, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, June 1975.

American Journal of Computational Linguistics *Microfiche 34 : 19*

A TUNEABLE PERFORMANCE GRAMMAR

JANE J. ROBINSON

Artificial Intelligence Center
Stanford Research Institute
Menlo Park, California 94025

ABSTRACT

This paper describes a tuneable performance grammar currently being developed for speech understanding. It shows how attributes of words are defined and propagated to successively larger phrases, how other attributes are acquired, how 'factors' reference them to help the parser choose among competing definitions in order to interpret the utterance correctly, and how these factors can easily be changed to adapt the grammar to other discourses and contexts. Factors that might be classified as 'syntactic' are emphasized, but the attributes they reference need not be, and seldom are, purely syntactic.

Acknowledgment

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and monitored by the U.S. Army Research Office under Contract No. DAHC04-75-C-0006.

A performance grammar (PG) defines the form and meaning of the kinds of utterances that occur in spontaneous dialog. When the definitions of the grammar provide information that helps a parser choose those rules most likely to lead to correct interpretations of utterances, the grammar is said to be 'tuned'. When the tuning is easily changed when the domain of discourse changes, the grammar is said to be 'tuneable'. The ability to tune a grammar is particularly important in speech understanding where the inherent uncertainty of the input causes false paths through the grammar to be multiplied.

This paper describes a tuneable PG being developed jointly by SRI and SDC for a computer-based speech understanding system. Its vocabulary and phrase types, selected from protocols, are appropriate for asking and answering questions about properties of submarines. The PG now defines over 70 word and phrase categories. Its scope extends far beyond syntax. A discourse component enables it to handle anaphora and ellipsis, as in: "What is the surface displacement of the Lafayette?.... What is its draft?", and "What is the length of the Lafayette?.... The Ethan Allen?" A semantics component defines a common meaning for paraphrases, as in "the speed of the Lafayette is 30 knots" and "the Lafayette has a speed of thirty knots". (See Walker et al., 1975; Paxton and Robinson, 1975; Hendrix, 1975; Deutsch, 1975.)

Each definition composing the PG has three parts. The first names a word category or a phrase category and provides a context-free production for its composition. The second part, called 'attributes', tells how to determine the properties of an instance of the category. Any definition can reference multiple sources of knowledge--acoustic, syntactic, semantic, discourse, or pragmatic--for information needed to determine attribute values. The third part, 'factors', defines scores for combinations of attributes, indicating how well they 'fit'. It is through factor scores that the grammar is tuned. The individual scores are combined into a composite score which is used by the parser to choose among competing parsings. A purported instance of the definition with a score of OUT for any factor is immediately eliminated; a low score may eliminate a parsing path; a high score enhances the priority of a parsing path that applies the definition[9].

Our mnemonic terms for factor scores are VERYGOOD, GOOD, OK, POOR, BAD, and OUT. These are estimates of likelihood. They are necessarily vague, because we are dealing with gradual phenomena and probabilistic tendencies. They mean something like "quite likely", "expected", "ordinary", "odd but possible", "unlikely--listen again", and "so special that we do not define it". Rigid, prescriptive judgments are avoided. Combining "foot" with "-s" as a plural noun is indeed wrong and therefore OUT. On the other hand, "fuel" does combine with plural "-s" with the specialized meaning "kinds of fuel". At present, "fuels", like "foots" is judged to be OUT for our language, but this judgment can easily be altered, if we find that our language users refer to kinds of fuel as "fuels".

Since factor scores can be changed without affecting the rest of the definition, the grammar is tuneable to different discourse domains and styles of speaking. Also, if one factor defines a low score for an instantiation, others may still raise the composite score. A statistically improbable phrase that makes sense and is uttered intelligibly should not be unduly difficult to recognize and interpret.

The rest of this paper examines sequences of definitions required for parsing and understanding a typical utterance. We begin with word definitions, and show how the attributes of words are propagated to successively larger phrases, how other attributes peculiar to higher-level phrases are added, and how factors reference them in tuning the grammar.

Preceding discourse and underlying semantic distinctions constrain the surface syntax of an utterance. Because superficial syntactic properties signal those constraints, it is often economical to use syntactic factors in order to disconfirm a wrong parsing path or confirm a correct one, avoiding calls on semantics, discourse, and acoustics for expensive in-depth evaluations. For example, if someone says "fuel supplies", we do not want the parser to explore in depth the application of rules that build a plural noun-phrase from "fuel s..." without considering an alternative definition in which "fuel" is a modifier of a countable nominal beginning with "s". To this end, we include a factor that checks the countableness of "fuel" by referencing a count/mass/unit (CMU) attribute, which is syntax oriented but essentially semantics based.

Examples of some useful syntax-oriented attributes defined for the word category N (noun stem) appear in (1) below. Every N has a value for the CMU attribute drawn from the set (COUNT MASS UNIT). Ns with the CMU value UNIT (such as "foot", "ton", "knot") combine easily with plural suffixes and number expressions (e.g., "two knots", "five feet"), but not so well with definite determiners ("those two knots"), or genitive suffixes ("the twenty knots' speed"). (Cf. "the Ethan Allen's speed".)

```
(1)    WORDS.DEF       N
         FUEL                      CMU = (MASS);
         FOOT                      CMU = (UNIT), PLSUFF = NO;
         LAFAYETTE                 CMU = (COUNT);
         SURFACE.DISPLACEMENT      CMU = (COUNT), RELN = T;
         TON                       CMU = (UNIT);
```

Like the CMU attribute, the RELN attribute is essentially semantic. It marks such words as "surface displacement", "speed", "length", and "draft" as special 'relational' noun words. Syntactically, relational Ns do not combine readily with plural suffixes and number expressions, and when they do, the meaning is specialized. To some degree, they are like mass Ns; "three speeds" (three rates of speed) is analogous to "three fuels" (three kinds of fuel). However, "a speed of twenty knots" is acceptable, while "a fuel of two tons" is ill formed.

The attribute PLSUFF distinguishes irregular plurals like "foot". Unlike the CMU and RELN attributes, it is purely syntactic.

Attributes affecting the ability to combine with the plural suffix "-s" are referenced in the two composition rules of (2), defining the category NOUN. The attribute statements propagate the attributes of the stem, adding a number attribute (NBR). The first factor of N1 references the CMU attribute and states that if the value is MASS, then the score is GOOD. This judgment incorporates our knowledge that the other rule, N2, cannot apply to mass noun-stems. If the token is a mass noun-stem, N1 is the right composition rule to apply.

```
(2)    RULE.DEF N1      NOUN = N;
           ATTRIBUTES
               CMU,RELN,PLSUFF FROM N, NBR = "(SG);
           FACTORS
               CMU = IF CMU EQ "(MASS) THEN GOOD ELSE OK,
               RELN = IF RELN EQ "T THEN GOOD ELSE OK,
               PLSUFF = IF PLSUFF EQ "NO THEN GOOD ELSE OK;
           EXAMPLES
               SURFACE DISPLACEMENT, FOOT. FUEL (GOOD)
               SUBMARINE (OK);

       RULE.DEF N2      NOUN = N -PL;
           ATTRIBUTES
               CMU,RELN,PLSUFF FROM N, NBR = "(PL);
           FACTORS
               PLSUFF = IF PLSUFF EQ "NO THEN OUT ELSE OK,
               CMU = IF CMU EQ "(MASS) THEN OUT ELSE OK,
               UNIT = IF "UNIT IN CMU THEN GOOD ELSE OK,
               RELN = IF RELN EQ "T THEN POOR ELSE OK;
           EXAMPLES
               FOOT -S, FUEL -S (OUT), TONS (GOOD)
               SURFACE DISPLACEMENTS (POOR), SUBMARINES (OK);
```

Like the CMU factors, the PLSUFF factors enhance the score for applying N1 to stems that do not take a plural suffix and constrain N2 not to apply. A RELN factor enhances the score when

N1 is applied to a relational noun stem, and lowers the score when N2 is applied. Plurals of such stems are less likely than singulars, but are possible. The UNIT factor of N2 enhances the score when the composition rule applies to an N with the value UNIT in its CMU attribute. This judgment is based on the fact that, in our current task domain, all the measured properties have measurements exceeding one unit and on the reasonable expectation that exactly one unit is a special case.

The attributes of the Ns continue to be propagated through successive composition rules so that noun phrases acquire the attributes, with some exceptions and additions, of the Ns that are their heads. One of the additional attributes of NPs is FOCUS: a noun phrase is definite (DEF) or indefinite (INDEF).

Combining definite focus with a unit is unusual. Compared with "which submarine", "which seven thousand tons" seems odd, as does "those twenty knots" and "a draft of the five feet". Indefinite focus is more common for units: "a ton", "a draft of five feet", "twenty knots". It does not suggest a uniquely determinable object or set of objects, pointed to in the discourse.

How these syntactic tendencies are handled in three composition-rule definitions is shown in (3). Factors in NP4 eliminate expressions like "five fuels", "five submerged speeds of three knots", "how much submarine", "one submarines" and "how many fuel". They score expressions like "five feet" as VERYGOOD and "five submarines" as OK. Factors for NP7 eliminate "those submarine", "those fuels" and accept "what fuel" as OK, while "which tons" and "that draft of five feet" are POOR. Factors for NP11 eliminate "a fuel", "a draft of the Lafayette", and "a submarines"; accept "a submarine", "a ton", "the submarine", and "the submerged speed", and score "the ton" and "the draft of five feet" as POOR.

```
(3)  RULE.DEF NP4    NP = NUMBERP NOM;
      ATTRIBUTES
            FOCUS = "INDEF, MOOD,NUM FROM NUMBERP, RELN FROM NOM,
            NBR = GINTERSECT(NBR(NUMBERP),NBR(NOM)),
            CMU = GINTERSECT(CMU(NUMBERP),CMU(NOM));
      FACTORS
            CMU = IF NULL CMU THEN OUT ELSE OK,
            HUN = IF FSTWD(NUMBERP) IN "(HUNDRED THOUSAND MILLION)
                THEN OUT ELSE OK,
            NBR = IF NULL NBR THEN OUT ELSE OK,
            UNIT = IF "UNIT IN CMU THEN VERYGOOD ELSE OK,
            RELN = IF RELN EQ T THEN OUT ELSE OK;

    RULE,DEF NP7    NP = DET NOM;
      ATTRIBUTES
            FOCUS = "DEF, RELN FROM NOM, MOOD FROM DET,
            CMU = GINTERSECT(CMU(DET),CMU(NOM)),
            NBR = GINTERSECT(NBR(DET),NBR(NOM));
      FACTORS
            CMU = IF NULL CMU THEN OUT ELSE OK,
            UNIT = IF "UNIT IN CMU THEN POOR ELSE OK,
            NBR - IF NULL NBR THEN OUT ELSE OK;

    RULE,DEF NP11    NP = ART NOM;
      ATTRIBUTES
            RELN FROM NOM, FOCUS FROM ART, MOOD = "DEC,
            CMU = GINTERSECT(CMU(ART),CMU(NOM)),
            NBR = GINTERSECT(NBR(ART),NBR(NOM));
      FACTORS
            CMU = IF NULL CMU THEN OUT ELSE OK,
            NBR = IF NULL NBR THEN OUT ELSE OK,
            UNIT = IF "UNIT IN CMU AND FOCUS EQ "DEF
                THEN POOR ELSE GOOD,
            RELN = IF RELN EQ T AND FOCUS EQ "INDEF AND
                CMU EQ "(COUNT) THEN OUT ELSE OK;
```

In each definition, a UNIT factor references the CMU attribute of the NP. If the value is NIL, the definition is not applicable. If UNIT is a value, then the UNIT factor for NP4 scores the application as VERYGOOD. There are two reasons for this judgment. Number expressions are typically found with unit expressions to form measure expressions, and units are more likely to occur with indefinite than with definite focus, as the preceding examples ("twenty knots" and so on) have indicated.

Since the focus for NP7 is always definite, the UNIT factor decreases the score for applying it when the UNIT value appears in the CMU attribute. For NP11, the UNIT factor scores the application GOOD if the article is "a" and UNIT appears in the CMU values, but POOR if the article is "the".

NP4 applies especially well to instances in which units are present, but does not apply at all if the head of the nominal constituent is a RELN stem. In discourse about washing machines and bicycles, "three speeds" might occur in an ordinary way, but for our current discourse, we do not anticipate such a combination. Certainly, we do not expect "three surface displacements".

Such constraints relieve the need for detailed analysis. For instance, assume that the acoustic mapper has tentatively offered both "submarine" and "submerged speed" as acoustically plausible alternatives for filling the gap in the partially analyzed phrase "three ----- -s of the U.S. Navy". This is not improbable since "submarines" and "submerged speeds" resemble each other in many ways. They both start with "s"; their first syllables have central vowels; their last syllables have high front vowels; and so forth. If NP4 is to be applied, however, the RELN factor will resolve the doubt in favor of "submarine", and there will be no need to test in depth how well "submerged speed" maps onto the acoustic data or fits the semantic and discourse constraints.

The UNIT factor of NP11 guides the choice between "a" and "the", where acoustic evidence for a choice is typically lacking. Semantically, "a" resembles "one" in its ability to combine with numbers and units; e.g., "one ton", "a ton", "one hundred", "a hundred". If the instance of the NOM is "ton", "foot", "knot", or some other singular expression with the value UNIT for its CMU attribute, then "a" is judged to be more likely than "the". On the other hand, if the NOM is "fuel" or "submarines", the article cannot be "a". The CMU attribute for "a" is (COUNT UNIT), which does not intersect with the value (MASS) of the CMU attribute for "fuel"; the NBR attribute is (SG), which does not intersect with the value (PL) for "submarines". The factors referencing these attributes rule out application when the intersection is NIL. These are typical syntactic agreement tests.

As longer phrases are built up, the various attributes interact in other ways. For instance, the syntactic properties of relational expressions depend on which aspect of the relation is present in an accompanying prepositional phrase. Prepositional phrases have the attributes of their NP objects. When a prepositional phrase modifies a noun with the RELN attribute, the CMU attribute for the resultant phrase is defined by taking the union of the values for the two nominal constituents. As a result, phrases like "surface displacement of the Lafayette" have the value (COUNT) and those like "surface displacment of seven thousand tons" have the value (COUNT UNIT). The difference in values marks the fact that the two examples do not fit witn equal ease in all syntactic environments. It is referenced in the UNIT and RELN factors in (3) above, to influence the choice between the two articles, which are seldom distinguished clearly by sound. The rule is tuned to prefer "the" in the absence of the UNIT value; as in "the surface displacement of the Lafayette", and "a" when it is present, as in "a surface displacement of seven thousand tons". "A surface displacment of the Lafayette", which implies the possibility of having more than one surface displacement, is ruled out completely.

NPs also have a MOOD attribute, derived from their initial constituents. It is either declarative (DEC) as in "this submarine", or WH-interrogative (WH) as in "which submarine". The WH value is propagated to the larger phrases in which NPs are constituents. Sentences (S) and utterances (U) take the value for their MOOD attribute from an initial NP. Our current vocabulary does not include verbs like "know" and "tell", which can embed WH questions like "Do you know what the surface displacement is?" For the time being, we assume that noninitial noun phrases are not likely to have the value WH for MOOD. Echo questions, e.g., "You said what?" are not ruled out, but have lower scores.

The convergence of many attributes at the higher levels of phrase composition makes possible many discriminatory judgments. Some of them are shown in (4).

```
(4)  RULE.DEF S3     S = NP:NP1 AUXB NP:NP2;

       ATTRIBUTES
        MOOD,FOCUS,CMU,RELN  FROM NP1,
        AFFNEG FROM AUXB;

       FACTORS
        NBRAGR1 = IF  CMU EQUAL "(UNIT) THEN
             [IF NBR(AUXB)EQUAL "(SG)THEN OK ELSE OUT]ELSE
              IF GINTERSECT(NBR(NP1),NBR(AUXB))THEN OK ELSE OUT,
        NBRAGR2 = IF  CMU(NP2) EQUAL "(UNIT)  THEN OK ELSE
            IF GINTERSECT(NBR(NP2),NBR(AUXB))THEN OK ELSE OUT,
        FOCUS = IF FOCUS(NP1) EQ "INDEF AND FOCUS(NP2) EQ "DEF
            THEN POOR ELSE OK,
        GCASE1 = IF GCASE(NP1) EQUAL "(ACC) THEN OUT ELSE OK,
        GCASE2 = IF GCASE(NP2) EQUAL "(ACC) THEN OUT ELSE OK,
        MOOD1 = IF MOOD EQUAL "(WH) THEN GOOD ELSE OK,
        MOOD2 = IF MOOD EQUAL "(WH) AND MOOD(NP2) EQUAL "(WH)
             THEN POOR ELSE OK,
        AFFNEG = IF MOOD EQUAL "(WH) AND AFFNEG EQ "NEG THEN
           BAD ELSE OK,
        RELN = IF RELN EQ "T AND CMU(NP2) EQUAL "(UNIT)
                THEN VERYGOOD ELSE OK,
        PERSAGR = IF GINTERSECT(PERS(NP1),PERS(AUXB))
                   THEN OK ELSE OUT;

       EXAMPLES
        THE LAFAYETTE IS A SUBMARINE (OK)
        THE LAFAYETTE IS SUBMARINES, WHAT IS THEM (OUT)
        A LAFAYETTE IS THE SUBMARINE (POOR)
        THEM ARE SUBMARINES, IT AM A SHIP (OUT)
        WHAT IS IT, WHAT IS THE LENGTH (GOOD)
        HOW MANY ARE WHAT (POOR)
        WHAT ISN'T THE SURFACE DISPLACEMENT (BAD)
        THE SURFACE DISPLACEMENT IS 7000 TONS (VERYGOOD);
```

The PERSAGR (person-agreement) factor tests for agreement between the so-called pronouns and the auxiliary constituent

The two grammatical case factors, GCASE1 and GCASE2, require that the grammatical cases of the two NPs are not accusative. These traditional syntactic agreement tests block application of the composition rule to putative expressions like "it are" and "they is". "Them is" is doubly blocked.

Some of the remaining factor statements in (4) are less traditional. One of these is the AFFNEG factor, which references both the MOOD and AFFNEG attributes and reduces the score greatly if the instance is purportedly a negative WH question like "what isn't the surface displacement?" Genuine requests for negative information occur in highly circumscribed situations. The rhetorical question is not a genuine request for information (e.g., "Who wouldn't like to be rich and famous!"). "Who isn't here?" is reasonable only if there is an established and limited list of people who are expected to be present, as in a classroom. "What isn't your name?" and "Where don't you live?" are patently absurd.

The constraint on negative WH questions is essentially due to pragmatic forces as well as semantic ones. Similar forces are at work in observed tendencies for the first NP in the composition defined by S3 to be indefinite in focus only when the second one is also. Stated oversimply, in coherent discourse, the things already talked about--the "old" information--tends to come first. What is predicated about it--the "new" information--tends to follow. Old information is information that has already been talked about and established in the discourse, so that it is likely to be encoded in definite noun phrases. These are likely to be in subject position, so that the sentence they introduce is consistent with preceding sentences. New information tends to be introduced in indefinite noun phrases. The next mention of the "same thing" will then be old information, eligible for definite focus. Consequently, "A Lafayette is that submarine" seems peculiar, relative to "That submarine is a Lafayette". "A Lafayette is it" is still more peculiar. These discourse-based probabilistic tendencies are expressed in the FOCUS factor of S3.

The CMU attribute, as previously noted, is not purely syntactic. On the other hand, matters like number agreement have always been central to syntax. It is particularly interesting, therefore, that the number agreement constraints for S3 cannot be properly stated without appealing to CMU. To state number agreement constraints, Ns denoting units must be marked separately. Sentences like "These are a submarine", "These is a torpedo tube", "These is missile launchers", and "This are subs" are clearly ungrammatical, and the ungrammaticality is usually attributed to the fact that one of the constituents differs in grammatical number from the other two. However, "The surface displacement is seven thousand tons" is wholly grammatical even though two of the constituents are singular and the third is plural. Such use of semantic attributes in syntactic factors points to the conclusion that the integration of information from different sources of knowledge is well motivated on both linguistic and heuristic grounds.

Because of the high frequency of WH questions in the protocols from which the vocabulary and phrase types were selected, the PG is now tuned to expect them. A sentence defined by S3 receives a higher score from the MOOD1 factor if its MOOD is WH. This tuning can easily be changed without altering the syntax or semantics of the language. If the user both extracts data from the data base and enters data into it, with no predictable pattern of alternation, factors like MOOD1 can simply be removed. A more interesting alternative is to reset them dynamically in a discourse context where the computer sometimes asks questions for the user to answer. After each user question, the grammar could be tuned to expect a declarative utterance whose syntax and semantics were appropriate and relevant.

References

Deutsch, B. G. Establishing Context in Task-Oriented Dialogs. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Hendrix, Gary G. Semantic Processing for Speech Understanding. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Paxton, William H., and Robinson, Ann E. System Integration and Control in a Speech Understanding System. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Walker, Donald E., et al. Speech Understanding Research. Annual Report, Project 3804, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, June 1975.

American Journal of Computational Linguistics *Microfiche 34 : 34*

SEMANTIC PROCESSING FOR SPEECH UNDERSTANDING

GARY G. HENDRIX

Artificial Intelligence Center
Stanford Research Institute
Menlo Park, California 94025

ABSTRACT

The semantic component of the speech understanding system being developed jointly by SRI and SDC rules out phrase combinations that are not meaningful and produces semantic interpretations for combinations that are. The system consists of a semantic network model and routines that interact with it. The net is partitioned into a set of hierarchically ordered subnets, facilitating the encoding of higher-order predicates and the maintenance of multiple parsing hypotheses. Composition routines, combining utterance components into phrases, consult network descriptions of prototype situations and surface-to-deep-case maps. Outputs from these routines are network fragments consisting of several subnets that in aggregate capture the interrelationships between a phrase's syntax and semantics.

Acknowledgement

This research was supported by the Defense Advance Research Projects Agency of the Department of Defense and monitored by the U.S. Army Research Office under contract No. DAHC04-75-C-0006.

OVERVIEW

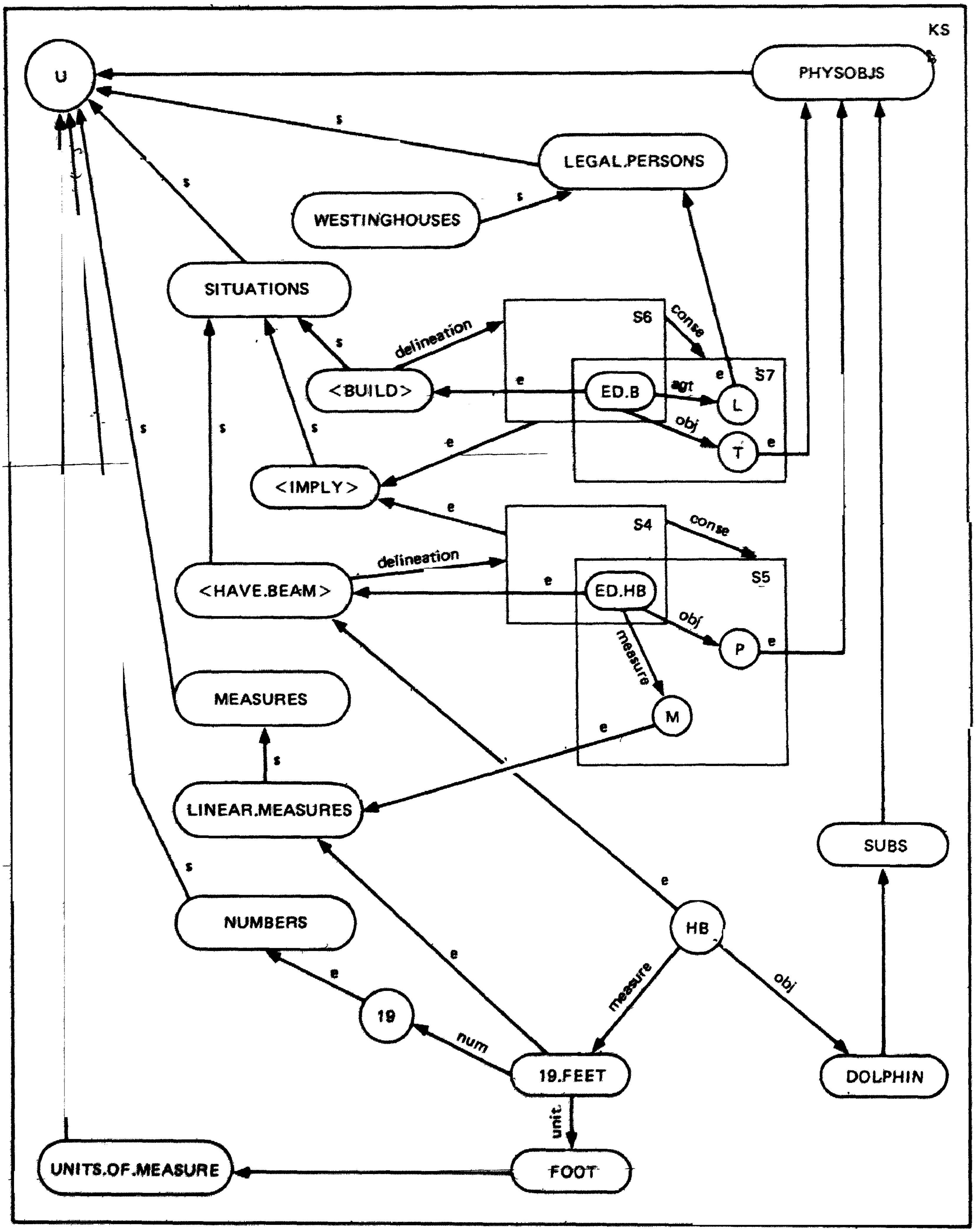
This paper describes aspects of the semantic component of the speech understanding system currently being developed jointly by SRI and SDC. (For a comprehensive discussion of nonacoustic portions of this system, see Walker et al., 1975.) The semantic component consists of two major parts: a semantic network coding a model of the task domain and a battery of semantic composition routines (SCRs) that are coordinated with the language definition (roughly, the "grammar" for the speech understanding system; see Paxton and Robinson, 1975, and Robinson, 1975). This paper concentrates exclusively on the interplay between these two major parts during parsing. However, the semantic component also plays important roles in knowledge management, discourse analysis, prediction, and question answering.

An SCR is called with network representations of components that the associated language definition rule has found to be syntactically capable of combining to form a larger phrase. Using knowledge from the semantic net, the SCRs eliminate combinations that, although syntactically acceptable, do not meet semantic criteria for meaningful unification. For combinations that are acceptable, the SCRs build network structures to represent the meaning of the composite phrase, using the network structures of the components as building blocks. These net structures are constructed so that (1) multiple hypotheses concerning the proper incorporation of a given utterance constituent in larger phrases may be encoded simultaneously in one net, (2) competing users of a constituent may share a single network structure representing the constituent, and (3) the association between each syntactic unit of an input and its translation image in the network is explicitly encoded for use in discourse analysis.

THE SEMANTIC NETWORK

The semantic network is the principal information source for SCRs, encoding such diverse entities as objects, situations, categories, taxonomies, definitions, and quantified statements. Network structures indicating possible relationships between objects are used to determine the meaningfulness of phrase combinations, while the network itself serves as the medium for recording interpretations of utterance fragments during parsing. The structure of this network differs from that of conventional nets in that nodes and arcs are partitioned into "spaces". These spaces, playing in networks a role roughly analogous to that played in strings by parentheses, group information into bundles that help to condense and organize the network's knowledge. An introduction to net partitioning is provided elsewhere (Hendrix, 1975).

An illustrative portion of the permanent knowledge section of the semantic network is depicted in Figure 1. In the upper left corner is node 'U', representing the universal set U. To the right is node 'PHYSOBJS', representing the set PHYSOBJS of



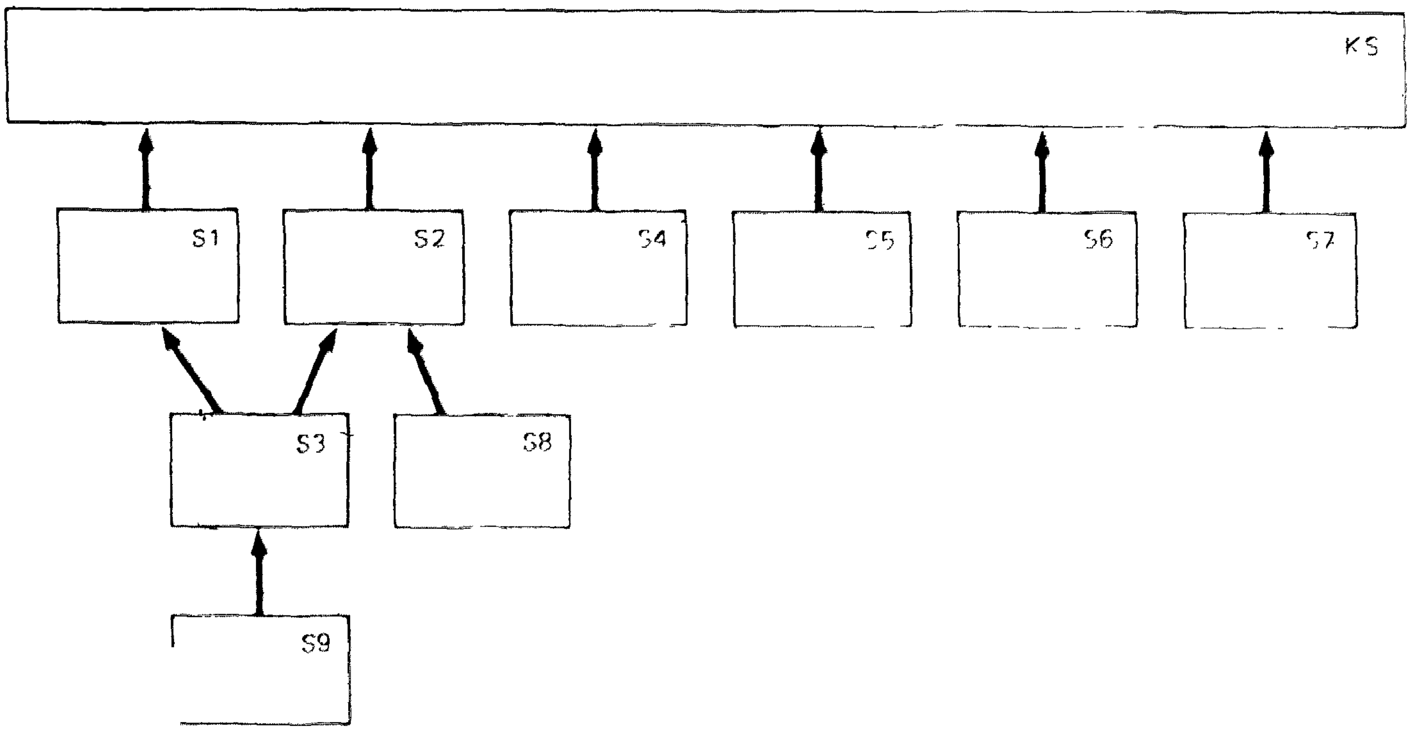
SA-3804-33

FIGURE 1 A SAMPLING FROM THE GENERAL KNOWLEDGE NET physical objects. That PHYSOBJS is a subset of U is indicated by the s-arc from 'PHYSOBJS' to 'U'. A subset of PHYSOBJS is SUBS, the set of all submarines. A particular element of SUBS, as indicated by the e-arc from 'DOLPHIN' to 'SUBS', is the DOLPHIN.

The DOLPHIN is a participant in a particular situation, HB, the situation in which the DOLPHIN has a beam of 19 feet. HB is an element of <HAVE,BEAM>, the set of all situations in which a physical object is characterized by a measure of its breadth. Certain outgoing arcs from a node representing a situation are used to specify situation attributes through deep semantic cases. For example, the outgoing obj-arc from 'HB' specifies the value of the "obj" (object) attribute of HB to be DOLPHIN. Hereafter the notation "#@obj" will be used to indicate "the value (#) of the attribute (@) obj."

The network of Figure 1 has been divided into five spaces, KS, S4, S5, S6, and S7. Pictorially, each of these spaces is represented by a box. The most global information in the network is encoded in space KS (the outermost box, sometimes called the "Knowledge Space") which includes such entities as nodes 'U' and 'PHYSOBJS' and the s-arc connecting them. The boxes representing spaces S4 through S7 may be thought of as holes in the box of KS. Paralleling the relationship between an inner and an outer block of an ALGOL program, each of these spaces specifies a more local area of the net than is specified by KS. From the perspective of S5, for example, it is possible to access both local node 'P' and (relatively) global node 'PHYSOBJS'. However, from KS the nodes and arcs inside S5 are not accessible. The hierarchy of space localization may be represented by a partial ordering such as that of Figure 2. From any space S, the nodes and arcs are accessible that lie in S or in any space S' above S in the hierarchy. For example, from S3 only nodes and arcs in S3, S2, S1, and KS are accessible.

Pictorially, it may be necessary to draw an arc crossing box boundaries. In such cases, the arc belongs to the space (or spaces) in whose box the arc label is written. Spaces may overlap. For example, in Figure 1, node 'ED,HB' lies in both space S4 and space S5. Further, a space may serve as a node in a more global space. Both S4 and S5 behave as nodes in KS and are connected by a conse-arc (consequence).



SA 3804 34

FIGURE 2 SPACE LOCALIZATION HIERARCHY

Typically, localized spaces such as S4 and S5 are used to encode higher-order "predicates," such as quantifiers, logical connectives, and hypothetical data. Here, S4 and S5 are used to encode an implication. The space S4, doubling as a node in space KS, is connected by an e-arc to '<IMPLY>' and by a conse-arc to 'S5'. The interpretation of any element of set <IMPLY> is that if entities can be found matching the structure of the element space, then the existence of entities matching the structure of the associated conse space may be inferred. The only structure encoded in element space S4 is a node ('ED.HB') with an e-arc to '<HAVE.BEAM>'. This structure matches any concrete instance of <HAVE.BEAM> (such as HB). Thus, for any instance of <HAVE.BEAM>, entities matching the structure of S5 must exist. The structure of S5 indicates that the element of <HAVE.BEAM> will have a #@obj, which is an element of PHYSOBJS, and a #@measure, which is an element of LINEAR.MEASURES.

The implication encoded by S4 and S5 serves to delineate the set <HAVE.BEAM>. That is, the implication indicates all the attributes (deep cases) of a <HAVE.BEAM> situation and their ranges of acceptable values. This delineation may be used during parsing to test the plausibility of a given group of entities being united in a <HAVE.BEAM> situation or, in a predictive mode, to suggest possible sentence participants. Such delineations are encoded for every situation and event set known to the system, a second example in Figure 1 being the delineation of set <BUILD>.

THE SYSTEM IN ACTION

The use of the SCRs and semantic network in translation may be seen by considering the parsing of

"The power plant of the sub was built by Westinghouse."

The ultimate result of the translation process for this utterance is the network structure recorded in the SCRATCH space of Figure 3. Structures representing new inputs are constructed in a scratch space (or spaces) to prevent them from becoming confused with the system's permanent knowledge (recorded in KS). Since the system understands new inputs by appealing to previous knowledge, there are many links, in the form of e-arcs, from the SCRATCH space into KS. (Note: Only a fragment of KS is shown in the various figures of this paper.)

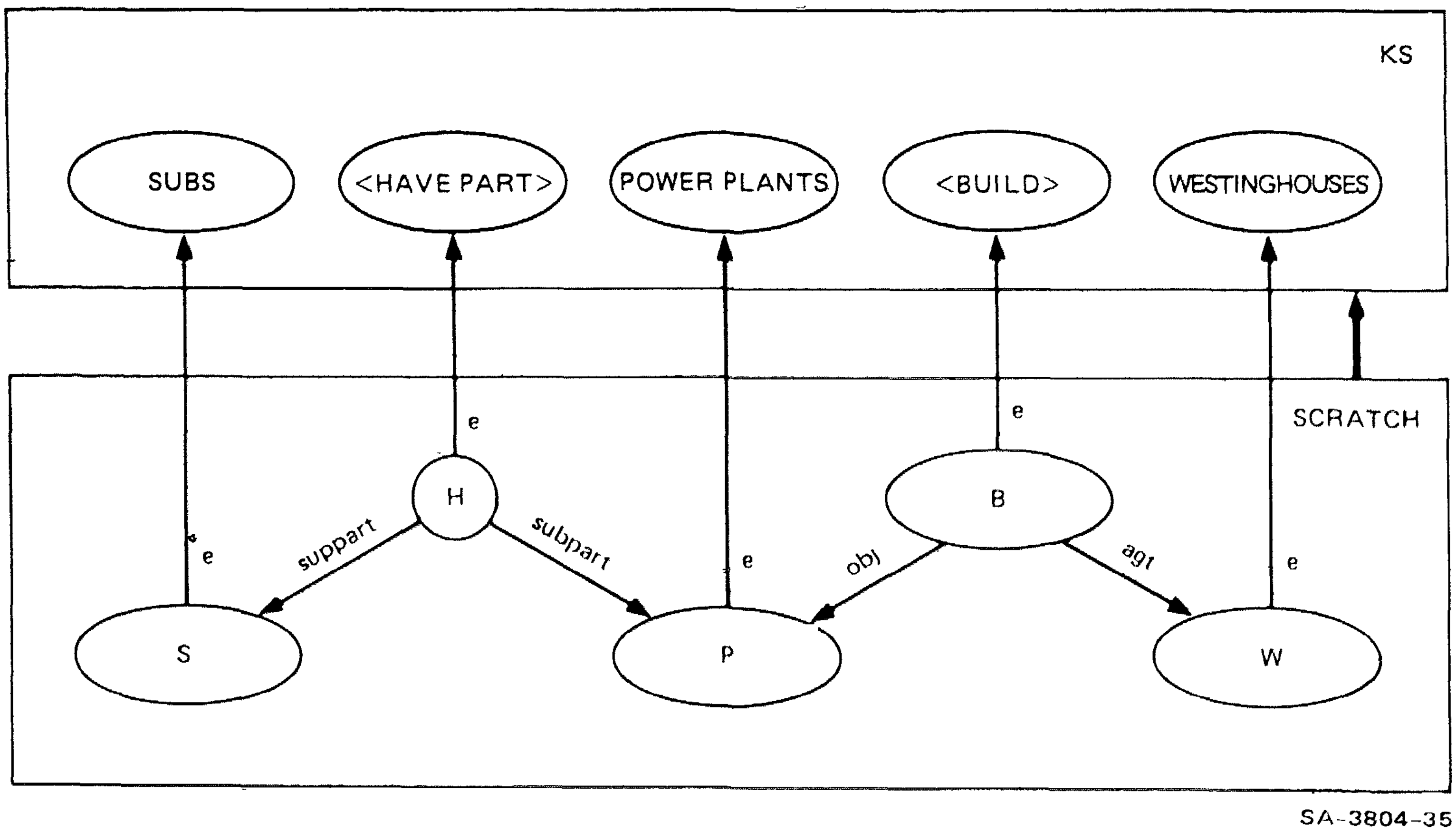


FIGURE 3 PARSE TARGET STRUCTURE FOR "THE-POWER-PLANT OF THE-SUB WAS-BUILT BY WESTINGHOUSE"

The interpretation of the network in the SCRATCH space is as follows: Node 'B' represents an element of the set <BUILD> of building events in which a #@agt W built a #@obj P. The agent W of the building event is an element of the set of WESTINGHOUSES. The #@obj built by W is P, an element of the set POWER.PLANTS. According to node 'H', this power plant is the #@subpart in a <HAVE.PART> relationship in which S, the particular member of SUBS currently in context, is the #@suppart (superpart). Discourse analysis mechanisms discussed in Deutsch (1975) and, more fully, in Walker et al. (1975) will be used to associate W with the unique Westinghouse Corporation known to the semantic net in space KS. The other definite NPs ("the sub" and "the power plant of the sub") will likewise be resolved.

To suppress secondary details while considering the building of this structure, assume the highly simplified language definition:

Grammar	Lexicon
R1: S => NP VP	NP: the-power-plant,
R2: NP => NP PREPP	the-sub, Westinghouse
R3: VP => VP PREPP	VP: was-built
R4: PREPP => PREP NP	PREP: of, by

(Note: "the-power-plant" is not treated as an NP in the actual system. Rather, NOM "power plant" is first combined with PREPP "of the sub" and only afterward is "the" appended to produce the NP "the power plant of the sub".)

In the translation process, spaces are created to represent the semantics of each grammatically defined constituent of the total utterance. These spaces are shown in Figure 4 with heavy arrows indicating the space hierarchy.

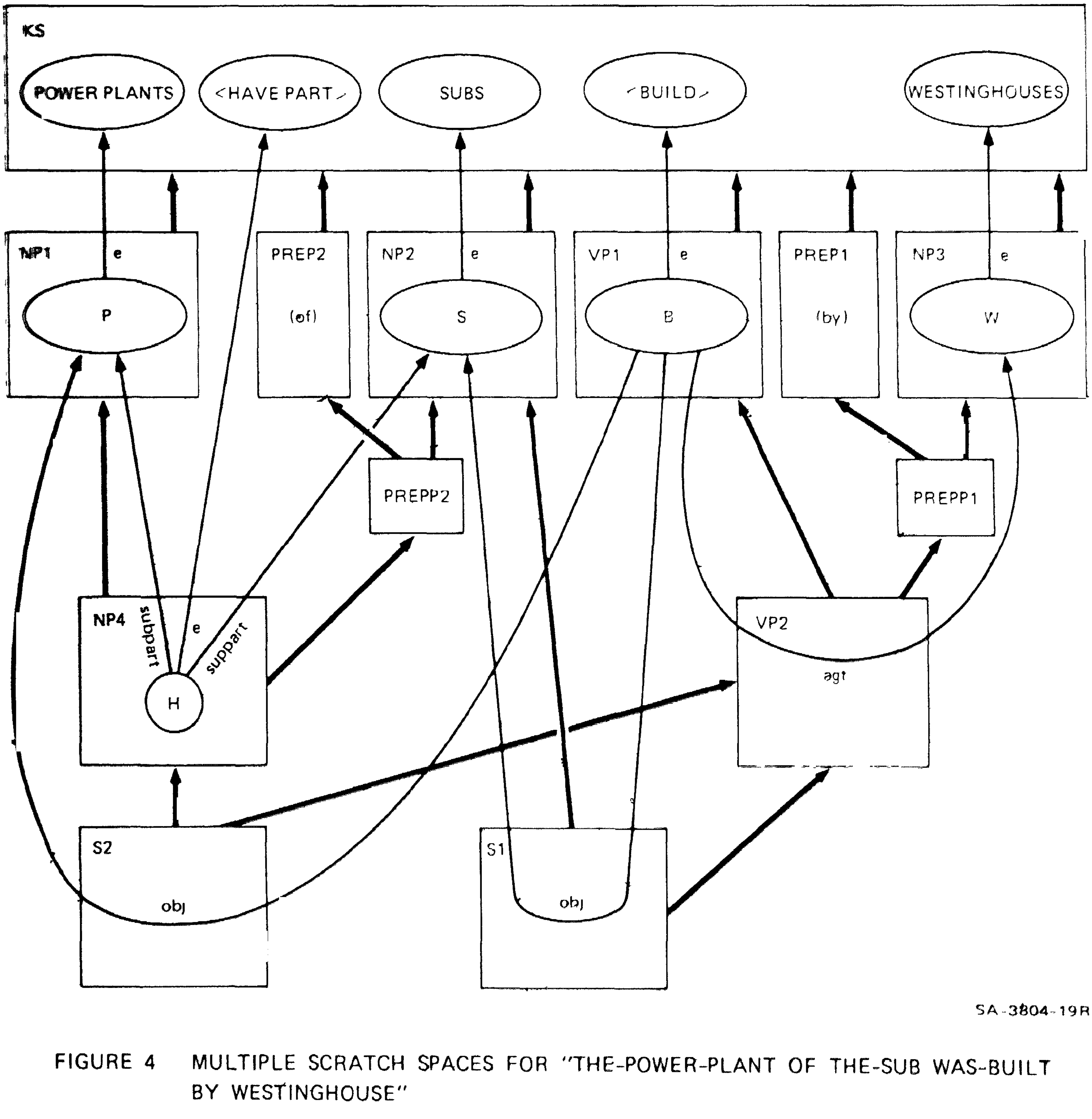


FIGURE 4 MULTIPLE SCRATCH SPACES FOR "THE-POWER-PLANT OF THE-SUB WAS-BUILT BY WESTINGHOUSE"

At the start of processing, space KS contains knowledge about power-plants, <HAVE,PART> relationships, submarines, <BUILD> events, and Westinghouse. On spotting the noun phrase "the-power-plant", an SCR is called to set up a space, NP1, below KS in the partial ordering. Within this space, a structure is created representing the meaning of "the-power-plant". Similarly, new spaces are set up to encode the other sentence constituents that correspond to explicit lexical entries.

As the parser groups subphrases into larger units, SCRs are called to aid in the process. Using rule R4, PREPP1 ("by") and NP3 ("Westinghouse") are combined to form PREPP1 ("by Westinghouse").[1] PREPP1 is allocated its own space, but no new structures are created within it.

When syntactic considerations suggests combining VP1 ("was-built") with PREPP1, the appropriate SCR is called. Consulting a surface-to-deep-case map associated with the lexical entry for the verb "build", the SCR determines that a "by" PREPP following the verb often signals the deep agt case in a passive construction. Operating under this hypothesis, the SCR checks the voice of VP1. Passing this test, the SCR next checks the semantic feasibility of the NP of PREPP1 serving as the #@agt in a <BUILD> event. To do this, the SCR consults the #@delineation of <BUILD> in space KS (see Figure 1). The delineation is encoded as an <IMPLY> situation in terms of spaces S6 and S7. As discussed earlier, this delineation indicates that any #@agt of a

<BUILD> situation must be an element of LEGAL.PERSONS. The candidate for the #@agt position is W of space NP3. Since W is an element of WESTINGHOUSES and WESTINGHOUSES is a subset of LEGAL.PERSONS, W is accepted. A construction such as "built by the submarine" would have been rejected.

Once VP1 and PREPP1 have passed the acceptability tests, a new space, VP2, is constructed to encode the resultant VP. This new space links node 'B' of VP1 with node 'W' of NP3 via an agt-arc. This new arc is accessible only from space VP2 (and lower spaces in the hierarchy) and is not accessible from either VP1 or NP3. This leaves the components encoded in VP1 and NP3 free to combine in alternatives to VP2 if need be.

Continuing the parse, NP2 ("the-sub") is combined with VP2 ("was-built by Westinghouse") to form S1, after passing tests similar to those above. The obj-arc linking the constituent phrases of S1 is contained in space S1 and hence is inaccessible from the spaces of the constituents. Notice that the construct "the-sub was-built by Westinghouse" which is encoded by S1 is a spurious interpretation of utterance components.

Using rule R4, PREP "of" may be combined with NP2 to form PREPP2. The network structures accessible from PREPP2 do not include the (spurious) obj-arc from 'B' to 'S' that lies in space S1.

When the syntax of rule R2 suggests combining NP1 and PREPP2 to form a new NP ("the-power-plant of the-sub"), an SCR is called. This SCR checks NP1 to see if it is relational in nature (as is "beam" in "beam of the Dolphin") and hence expecting an argument to be supplied. Since NP1 fails this test, the SCR checks the properties of the PREP "of" and discovers that it may be used to encode <HAVE.PART> situations. Calling upon the delineation of <HAVE.PART> and appropriate surface-to-deep-case maps, the SCR determines this to be a legitimate interpretation and hence builds space NP4 with a node 'H' and three arcs as shown. While these new constructs are accessible from space NP4, they are inaccessible from constituents NP1 and PREPP2 (and NP2). Furthermore, they cannot be accessed from spurious space S1; hence the construction of NP4 has not altered the view of the net from S1.

Using rule R1, S2 is constructed from NP4 and VP2. In addition to the obj-arc contained in space S2 itself, the view of the net from S2 includes all the information accessible from either space NP4 or space VP2 and hence is identical to the view from space SCRATCH of Figure 3. Since the parse corresponding to space S1 does not successfully account for the fragment "the-power-plant of", it is rejected, and S2 is accepted as expressing the meaning of the input.

The partial ordering of spaces from S2 to KS indicated in Figure 4 is identical to that represented more clearly in Figure 5, which, because of the choice of space labels, may be recognized as the parse tree of the input sentence. The syntax of the input and the association between each syntactic unit and its corresponding semantics has therefore been captured in the structures built by the SCRs.

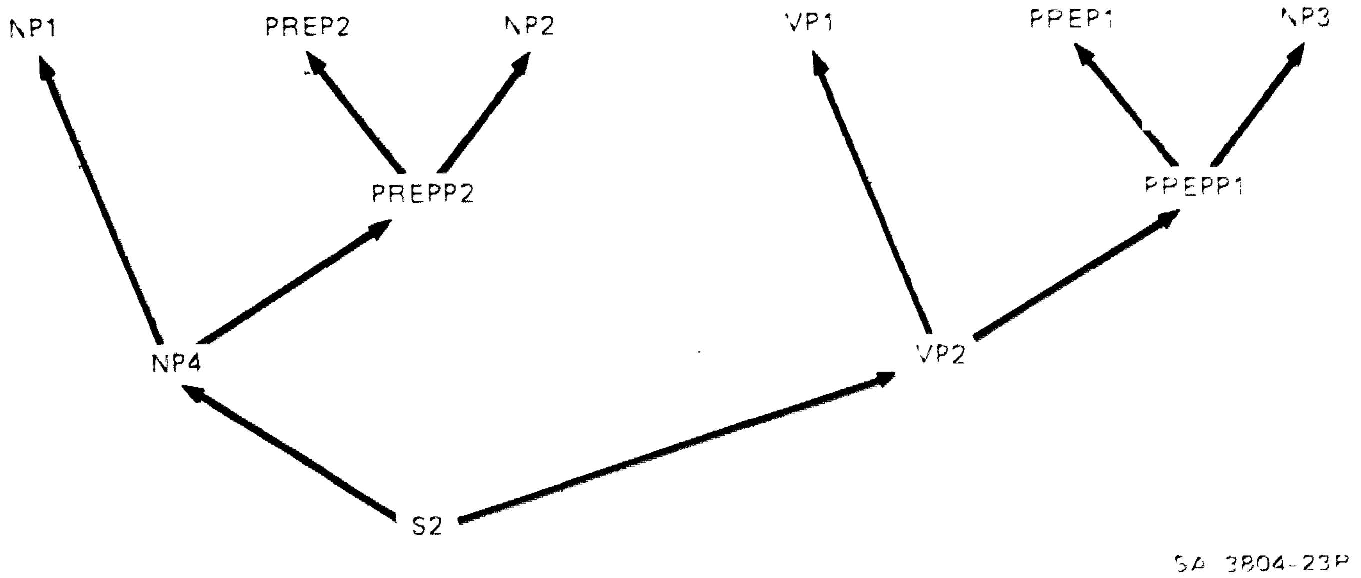


FIGURE 5 SPACE HIERARCHY ABOVE S2

DISCUSSION

Partitioning is a recent innovation in semantic networks. As shown above, this new feature enables networks to maintain alternative hypotheses (e.g., S1 and S2) concerning the use of utterance constituents and enables such competing hypotheses to share network subparts (e.g., VP2). Without partitioning, the back-linked nature of networks causes a constituent to be altered when it is incorporated into a larger unit and hence renders it unusable in alternative constructions. The highly ambiguous nature of acoustic input makes these abilities to maintain alternative hypotheses and share substructures especially important in speech understanding.

Partitioning also allows selected portions of a network to be associated with syntactic units, showing the correspondence between network entities and the syntactic structures that were used to communicate them. As discussed in the section on "Discourse Analysis and Pragmatics" in Walker et al. (1975), this association is crucial in analyzing the elliptic utterances that are so characteristic of speech.

References

Deutsch, Barbara G. Establishing Context in Task-Oriented Dialogs. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Hendrix, Gary G. Expanding the Utility of Semantic Networks Through Partitioning. Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, 3-8 September 1975, 115-121.

Paxton, William H., and Robinson, Ann E. System Integration and Control for Speech Understanding. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Robinson, Jane J. A Tuneable Performance Grammar. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Walker, Donald E., et al. Speech Understanding Research. Annual Report, Project 3804, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, June 1975.

American Journal of Computational Linguistics Microfiche 34 : 49

S.P S : A FORMALISM FOR SEMANTIC INTERPRETATION AND ITS USE IN PROCESSING PREPOSITIONS THAT REFERENCE SPACE

NORMAN K. SONDHEIMER AND DOYT PERRY

Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210

Introduction. This paper presents a formalism called Semantic Processing Scheme, SPS, for use in describing semantic interpreters. SPS is a rule-based system with a rule-ordering scheme that can produce deep case structures from phrase-structure trees. It was originally developed to demonstrate how English prepositions, such as "up", "down", and "through", which reference location, motion, and orientation in space could be semantically interpreted. This paper presents SPS in its current form and shows how it can handle these prepositions, called the locative prepositions. SPS is continuing to be used in studies of semantic processing.

Computational linguistics has seen a considerable amount of work on the development of general models for language-understanding systems. Among the most well-known examples of this is the work of Schank[4], Simmons[5], Winograd[7], and Woods[8,9]. On the whole, these models have been tested on broad but shallow subsets of English, in that they have been applied to many different phenomena but few extensively. The authors of this paper are taking a different approach. We are studying a few phenomena and attempting to allow for them in considerable detail. At the least, this approach should lead to better treatment of the particular phenomenon. It can also lead to the development of new general models or the revision of old ones.

The paper is written in five sections. The first describes the overall interpretative framework. A second indicates some of the difficulties inherent in the processing of locative prepositions. An overview of SPS is given in the third section. The last two sections expand on the SPS description and discuss how the locatives are allowed for.

Syntax, Semantics, and Pragmatics. SPS is developed for a traditional three-level system, with syntactic, semantic, and pragmatic stages. Based on the level of abstractness, these stages compare most closely to Winograd's and Woods'.

The syntactic processing stage is assumed to take strings of text and produce underlying syntactic structures in the form of constituent structure trees. We are attempting to keep these as close to surface constituent structures as possible. However, some divergence from the surface form is currently assumed. For example, imperatives, interrogatives, and relative clauses are assumed to be shown in a declarative-like form, and prepositions are assumed to have their complement immediately following them.

An SPS based interpreter takes these syntactic structures and produces output which reflects underlying semantic structures. The form of the semantic structures is also a topic of our research. We are using Case structures[1,2,4,5] and Planner-like assertional forms[7]. It is interesting to note that our results to date tend to indicate the need for a level of abstraction somewhere between Simmon's and Schank's semantic nets.

In developing the semantic level, we are trying to make it the one where "general knowledge of language and its relation to the world" is applied. This is in contrast to the pragmatic level, where situation-specific information is used to interpret the semantic structures.

In summary, a system employing SPS would construct syntactic trees, use SPS for the production of Case structures, and employ a pragmatic processing scheme to interpret these structures.

Problems in Processing Locative Prepositions. Part of the problem with the semantic interpretation of locatives is the complexity of the structures necessary to represent them on the underlying syntactic and semantic levels. This section discusses these problems and introduces our semantic structure notation.

The representation of locative prepositional meaning in Case structures has been problematic. The number of cases that Fillmore has postulated for them has risen to four--Location, Source, Goal, Path. He also features locatives in a paper on problems within Case grammar[2]. The worst of the problems involves not being able to interpret the semantic weight or meaning of the representation. An example of such a problem comes in the representation of the following: "Bill held his daughter on his lap in the tunnel.". Both of the locative phrases would be assigned the same case - Location. However, they actually locate different objects. Bill's daughter was said to be on his lap while both of them were said to be in the tunnel. Similarly, the use of an unordered set of cases fails to allow for the difference in meaning of the following two sentences, where the first two prepositional phrases in each would be in the Path case: "He went down the hill across the bridge to the chapel.", and "He went across the bridge down the hill to the chapel.".

The Case representation we are using deals with these problems. This representation uses only one case for all spatial references. This case, the <u>Place case</u>, identifies spaces which derive from the location of participants in its action, event, or state of affairs (or <u>event/state</u>). Which participants and how each space relates to them depends on the type of event/state.

The basic structure of the assertional notation can be seen by showing how a Place case would be represented: (:PLACE #E/S $PO). The ":" identifies a relation, the "#" an event/state, and the "$" objects (note that many of these will be replaced by variables in the actual assertions produced). The first element of any assertion is always a relation, which forces interpretations on the other elements. With the relation :PLACE, the last two elements must be references to an event/state and a spatial object (space), in that order. The specific spatial objects that are referred in Place assertions are called <u>Place objects</u>.

The prepositional elements on the semantic level can relate Place objects directly. An example of this is the representation of "She died away from where she lived.", i.e., (:PLACE #E/S1 $PO1)(:AWAYFROM $PO1 $PO2) (:PLACE #E/S2 $PO2). Here a prepositional element relates the Place object of the two event/states corresponding to "she died" and "she lived".

Prepositional elements can also relate spaces derived from Place objects. This is seen with the representation of motional meanings, such as in the multiple Path sentences above. The Place object of "go" and other motional event/states are taken as indicating the space traversed by the moving object or objects. For the example sentence, the Place object would show the space through which the person travelled. This is acceptable since the static positioning of these spaces (or paths) as "across" the bridge is logically equivalent to his going across it. The predication of derived spaces arises in the handling of the ordering problem. The motional Place object can be taken as composed of parts that are ordered like the parts of other objects (from front to back or top to bottom). The ordering here is based on the time the component spaces were occupied. Using relations to select segments of the path and the end points of these segments, simple mathematical relations compare the ordering of the component spaces, comparing parts of the journey in time. A semantic structure might look like the following: (:PLACE #X108 $X109) (:SEG $X109 $X110) (:SEG $X109 $X111) (:FINAL $X110 $X112) (:INITIAL $X111 $X113) (:LE $X112 $X113).

The Place case proposal avoids problems like that with the Location case example, through the representation of certain syntactically simple clauses with more than one event/state. The representation of "He held her on his lap in the tunnel." shows an event/state corresponding to "he held her" and one corresponding to "she was on his lap". These are constituents in a causative event/state, with the first causing the second

---

*Fillmore moves in this direction in [2]. Similarly, the representation resembles those of Rumelhart and Norman [3] and Schank[4]. We have attempted to systematically work out the event/state analysis, as far as it concerns locatives, for all verbs taking locative objects.[6]

This complex structure solves the case problems by allowing each preposition to predicate a different Place object. "On his lap" predicates an existential event/state showing where the female was located. "In the tunnel" can predicate the Place object of the causative event/state. The interpretation that space is that it is composed from the Place objects of its two constituent event/states. Hence, both people will be predicated by it.

While these last two devices enable us to avoid representational problems, it should, of course, be remembered that semantic interpretation must support these forms.*

Tied in with semantic complexity is also complexity on the syntactic level. Assuming sentences are normalized in underlying syntactic structures as specified, locatives appear in four positions: as the qualifier of a head noun in a noun phrase; as the complement of a copula; as the adjunct to a clause; and inside a clause as a locative object. The adjunct usage can be differentiated from the locative object by its tendency to give overall predication to the event or state referenced by the clause. In "He held her on his lap in the tunnel.", the first phrase is a locative object and the second is an adjunct.

To summarize this section has presented a variety of points about the semantic interpretation of locative prepositions- that they can require complex case representations, and that they appear in a variety of syntactic environments. SPS has been designed to relate the syntactic to the semantic

---

*There are other phenomena for which the Place case proposal allows. The complete representation is described elsewhere.[6] What has been given here is enough to show the difficulty of interpretation.

environment of locative prepositions. How it deals with these problems will be described after a brief overview of the formalism.

SPS. The SPS formalism is most closely related to a family of semantic interpretation schemes deriving from Woods' 1968 work.[8] The close similarity to that work lies in the basic form of rules. These rules have the form "pattern ⇒ action", where the pattern side specifies tests to be made on the syntactic structures, and the action side specifies forms to be added to the semantic structures. The tests are mainly based on the matching of tree fragments against syntactic structures and the testing of semantic features associated with those elements matched. In SPS, sets of features can be directly examined or compared to other sets of features. Each lexical entry may have multiple sets of features associated with it. SPS also allows these tests to be made against features associated with registers by other rules.

If the tests are successful, the action element is executed. This principally adds assertional forms to the semantic structure, but can also set values of registers. In the assertional forms, means are provided to allow references to the syntactic constituents and lexical entries matched, as well as to other forms through the registers.

SPS uses a finite state transition net for ordering the application of rules. Each noun phrase and sentence is analyzed under the control of a net associated with it. The process of forcing interpretation through constituents is guided by marking completely interpreted nodes. The overall tree is processed from the bottom up.

SPS Rules and Locative Prepositions. To see how SPS works in detail, and to explain how it allows for locative prepositions we look at a typical rule:

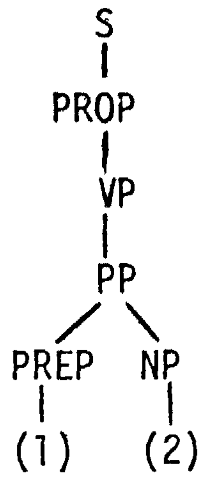
```
Rule 2-STAT-LO:
    ((*1-S5  (1 2 3 4)  I(4)  *1-S7
     (( EQ #2STAT 1-1)  (COMPATIBLE 1-1 2-1)
      (COMPATIBLE 1-2 OBJ(1-1)) (COMPATIBLE R(SS) SUBJ(1-1))))
   ⟹
    (((:PLACE R(CAUSED) !X(1)) (1-1 !X(1)  !X(2))
      (:PRED !X(3) $BE) (:OBJ !X(3) !1-2) (:PLACE !X(3) !X(2)))))
```

This is a rule that might be applied to interpret the prepositional phrase in the sentence "He held her on his lap.". The rule is identified as 2-STAT-LO. This particular name indicates that it deals with a preposition with a certain static type of meaning (2-STAT) used as a locative object (LO).

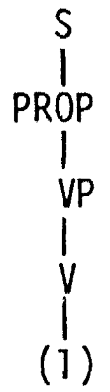
The pattern portion of the rule consists of two parts. The first describes the syntactic environment in which it applies, while the second gives the semantic feature tests.

The specification of the syntactic environment is done through reference to tree fragments that must be matched in the syntactic structure in order for the rule to apply. The reference is made through the asterisk-number-dash-literal forms in the rule, e.g., "*1-S5", where the literals identify fragments such as the following:

S5:



S7:



PROP = proposition

These fragments would match a locative object use of a preposition and the verb of that sentence. Other fragments are needed for other usages. The two forms in the rule after the reference to the first tree fragment will be described in the next section.

The second part of the pattern side is a set of triples used to test semantic features. These tests are of two types, EQ and COMPATIBLE. The EQ or "equal" tests ascertain the presence of a single feature in a set. Its first parameter is the feature and its second the set. The primary use of this test with locatives is to identify the cases where the prepositional tree fragment has actually matched a locative use of a preposition, since the syntactic parser can only be assumed to identify prepositions and not differentiate their senses. SPS allows for this discrimination by providing reference to the lexical entries associated with a preposition.* These references are made through the number-dash-number forms where the first number refers to the number associated with an occurrence of a tree fragment in a rule, while the second refers to the leaf number in the fragment.

The COMPATIBLE test is meant to allow for the semantic co-occurrence restrictions. It takes two sets of features as arguments and evaluates to true if the sets share at least one element. The above rule illustrates how this test can be used to allow for three types of restrictions affecting locatives. These are between a verb and its prepositional object and between a preposition and the two elements it relates (Winograd's semantic subject and semantic object).

The fact that SPS allows three sets of features to be associated with lexical entries is used for the three restrictions on locatives. One set, accessed through number-dash-number, is for restrictions placed on the

---

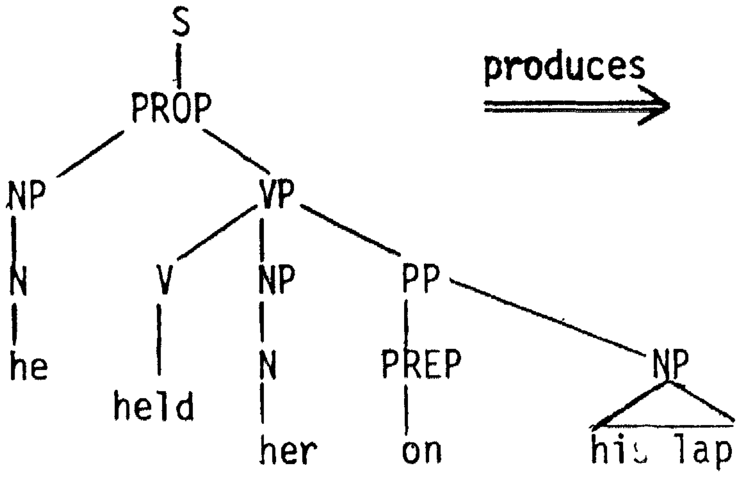
*With ambiguous entries, SPS tests each sense individually, therefore, any of the lexical references can be considered to have a unique meaning at any one time.

preposition by the verb. The other two sets, identified by the OBJ and SUB prefixes are for restrictions on the elements related.*

The final triple in the pattern differs from the others in that the test is against a register. SPS allows for registers that can have sets of features associated with them. The registers provide communication between rules to allow for some contextual effects. Tests may be made against registers both before and after they are set, with the test held in abeyance in the former case.

The use of the register here is to identify the semantic subject of the preposition. This is necessary since it can not be immediately said where the subject is situated in the sentence. In the following sentences it is initial, median, and final: "He held onto the rope.", "He held her on his lap.", and "He held in his hands the letter I sent Mary."

Given that everything is successful on the pattern side, the action side is executed. An example of rule application is given below:



```
(:PLACE !X21  !X100)
(:ON    !X100 !X101)
(:PRED  !X102 $BE)
(:OBJ   !X102 !X4)
(:PLACE !X102 !X101)
```

Note that ":PRED" identifies the predicator of an event/state, ":OBJ" identifies the element in the object case, and that the literals beginning with "!" are

---

*Note that the test using OBJ is on a noun phrase. At the moment SPS takes references to noun phrases and sentences to be to the lexical entries of their head noun and verb, respectively.

variables representing some event/states or objects. The purpose of the rule is to relate the location of the object being held to the location of the complement. These locations are available through event/states which identify where each of the two objects were. We use the predicator $BE for these event/states, such as in the one for "his lap" which is produced by the rule. How the correct assertions are produced from the assertional forms is illustrated in the above rule.

All the direct references to relations and objects that start with ":" "#", or "$" are inserted directly. The number-dash-number forms provide a reference to a literal stored in a lexical entry . For prepositions this literal gives the physical relation that the term refers to.

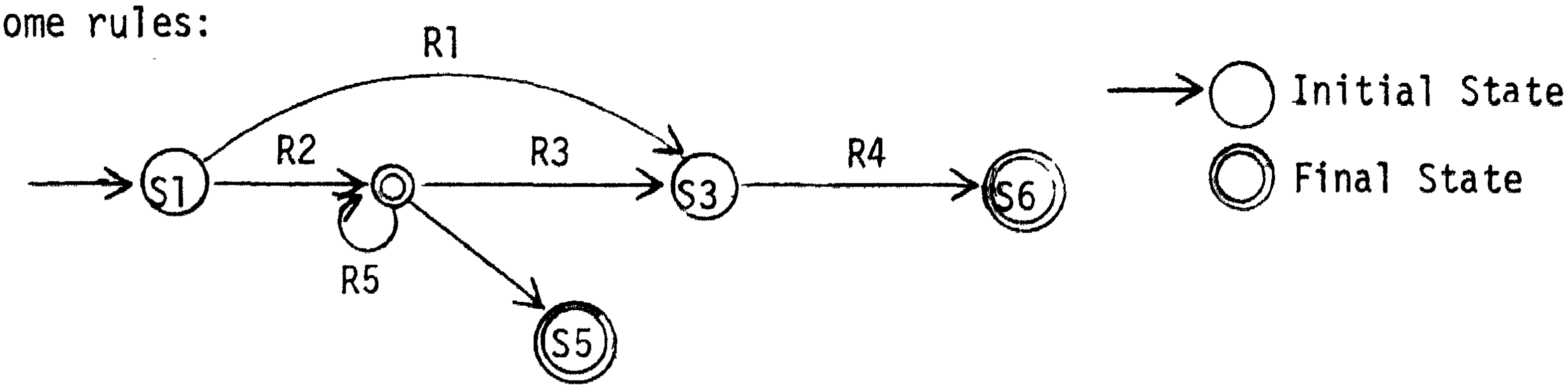
The two Place objects are formed by the use of a variable generation feature using the "!X("-number-")" form. References to the $BE event/state are also formed in this way. The other event/state is referenced through a register. SPS allows registers to hold variable names as well as feature sets. The register used here must be set with the variable name used when the event/state was constructed.

As the above example shows, the registers are used here in situations where more than one event/state results from a clause. When only one event/state exists, a simple reference to the major constituents of a sentence is necessary. SPS allows for this by automatically associating variables with the S and NP nodes in trees. These are referenced through forms like "!1-2" which here gets the variable associated with "his lap" (presumably !X4). This variable will also appear in the assertions describing the object; hence co-reference is achieved.

A facility of SPS missing from the example is register setting. Two operations can be accomplished. Either a variable is loaded, or both a variable and lexical entry are loaded.

These registers are essential to the development of the complex structures that must be produced at the semantic level. Besides helping produce multiple event/state structures, they also provide the means for ordering the partial predication of a path. In any list, the variable identifying the location of the last mentioned space can be loaded in a register. Then with the next phrase on the list, the variable can be referenced to form the comparison. The new final value can then be loaded in the register.

The Ordering of Rules and Locative Prepositions. The SPS system applies its rules in a strictly ordered fashion. Major constituents have rules applied to them on the basis of an ordering shown by a finite state transition network. The following is a hypothetical network for ordering the application of some rules:



The literals on the arcs name rules that must be successfully applied before a state change can occur. These nets are set up for noun phrase and sentential elements, and are used with a marking scheme such that interpretation of a constituent is complete only when its net is in a final state and all its constituents are marked as interpreted

These nets are set up for each head noun or verb to interpret noun phrases and sentences. Their utility is in allowing for the orderings among case elements. The constituents filling semantic roles in sentences can only appear in certain positions with respect to each other. This is particularly true with respect to verbs since the roles and orders differ from verb to verb. Hence, the net used depends on the head noun or verb.

There would be no need for a net if the number of constituents were strictly limited. However, with locatives there can be no limit on the number of intermediate points or on the successively finer specification of location, e.g., "He lives in New York near the Battery by a park...". Nets, with their ability to loop, are useful for these structures.

Interpretation proceeds from state to state until success or inability to progress further. In the latter case, SPS can back up to the last state that still had rules to apply, a fact useful in allowing for sematnic ambiguity.

Register tests have been mentioned as being postponed until the register is set. It could happen that the register never gets set, e.g., "He hits into the stands." does not specify what went into the stands. This is a case of semantic ellipsis. SPS allows default conditions to be associated with registers that are left tested but unset.

The means of progressing through a constituent and assuring its complete interpretation is provided by forced anchoring and marking schemes embedded in the rules. An example of each is seen in the rule shown in the previous section, i.e., "*1-S5 (1 2 3 4) I(4)". Both schemes refer to nodes in the tree fragments using a preorder - root first, then subtrees left to right. The numbers in the parentheses in the example rule refer to nodes of S5. The anchoring scheme restricts these nodes to being matched to the leftmost uninterpreted nodes in the structure being processed. When a node is prefixed by "I", it and the nodes it dominates are marked if the rule succeeds. Hence, the example rule marks the prepositional phrase as interpreted. Because of this marking scheme the noun phrases and sentences of a tree are interpreted from the bottom up.*

Conclusion. A formalism for writing semantic interpreters, SPS, has been described. It allows for a semantic feature scheme that can describe the restrictions on locative prepositions. SPS also has registers that can be used for these restrictions and for building up the case structures that represent the meanings of locatives. A rule-ordering scheme is also helpful here. It can be said that SPS is a good vehicle for interpreting locative prepositions, and that any system for semantic interpretation with these features will be able to analyze locatives. We do not claim that SPS is a completely successful semantic interpreter. However, the formalism seems to be clear and expressive and it does work for locative prepositions which, to the authors' knowledge, have not been as effectively dealt with elsewhere. It could well provide the basis for a uniform, coherent structure for semantic interpretation, especially for Case analysis. The authors intend to continue to experiment and develop it as a tool for language understanding.

SPS is implemented in LISP 1.6 on the DECSystem 10.

---

*More detail on a somewhat earlier version of SPS can be found in Chapter VII of [6].

Acknowledgements. Thanks are due to David Brown and Roberto Pardo for their reading of the paper and early programming work, respectively. The guidance of Richard L. Venezky is also gratefully acknowledged.

References.

1. Fillmore, Charles J., "The Case for Case" in *Universals in Linguistic Theory*, eds., Emmon Bach and Robert T. Harms, New York: Holt Rinehart and Winston, 1968, 1-88.
2. Fillmore, Charles J., "Some Problems for Case Grammar", in *Working Papers in Linguistics No. 10*, 245-265, Columbus: Department of Linguistics, The Ohio State University, 1971.
3. Rumelhart, David E. and Donald A. Norman, "Active Semantic Networks as a Model of Human Memory", in *Third International Joint Conference on Artificial Intelligence, Advance Papers of the Conference*, 450-457, Stanford, California, 1973.
4. Schank, Roger C., "Identification of Conceptualizations Underlying Natural Language", in *Computer Models of Thought and Language*, eds., Roger C. Schank and Kenneth M. Colby, 187-247, San Francisco: W. H Freeman and Company, 1973.
5. Simmons, Robert F., "Semantic Networks: Their Computation and Use for Understanding English Sentences", in *Computer Models of Thought and Language*, eds., Roger C. Schank and Kenneth M. Colby, 63-113, San Francisco: W. H. Freeman and Company, 1973.
6. Sondheimer, Norman K., "The Computational Semantics of Locative Prepositions", Unpublished Ph.D. Thesis, The University of Wisconsin, Madison, 1975.
7. Winograd, Terry, *Understanding Natural Language*, New York: Academic Press, 1972.
8. Woods, William A., "Procedural Semantics for a Question-Answering Machine", *AFIPS Conference Proceedings*, 33, (1968), 457-471.
9. Woods, William A., "Progress in Natural Language Understanding - An Application to Lunar Geology", *AFIPS Conference Proceedings*, 42, (1973), 441-450

American Journal of Computational Linguistics *Microfiche 34 : 64*

THE NATURE AND COMPUTATIONAL USE OF A MEANING REPRESENTATION FOR WORD CONCEPTS

NICK CERCONE
Department of Mathematical and Computing Sciences
Old Dominion University, Norfolk, Virginia 23508

ABSTRACT

Various representations have been used to portray the meanings of word (notably action) concepts. The most prominent of these include decomposition trees, linear representations such as the Predicate Calculus, and semantic networks. The proposition-based semantic network notation developed by Schubert (1974) is especially well suited for including pragmatic and semantic information as part of the meaning representation of individual word concepts. The attempt is made in this paper to explore the nature of word concepts whose meanings are represented as semantic networks and to investigate their computational use within the framework of a natural language processing system.

I. INTRODUCTION

The meaning of a concept is explained in terms of other concepts and throug its relationship to other concepts. Various representations have been used to portray the meanings of concepts. The most prominent of these include decomposition trees (Lakoff, 1972; Wilks, 1973), linear representations such as the Pre dicate Calculus (Sandewall, 1971), and semantic networks.

Natural language processing systems can conveniently utilize <u>factual</u> <u>knowledge</u> represented in the form of semantic networks. The visual suggestiveness of semantic networks aids both in the formulation and exposition of the computer data structures they resemble. The use of semantic networks can be found in the works of many authors writing on natural language processing (including Schank 1972, 1973; Quillian 1968, 1969; Simmons and Bruce 1971; Rumelhart et al. 1973; Anderson and Bower 1973; and Palme 1971) as well as other forms of understanding (including Winston 1970; and Guzman 1971).

In utilizing semantic network representations, these authors have made use of the following characteristics of semantic nets. First (and most important), nodes that denote the same concept are not duplicated (in most cases). It is then possible that distinct propositions may impinge on a node via arcs. Second, propositions are formed by linking predicate names to their argument nodes using arcs. Third, since concepts are not necessarily word concepts, particular and general concepts are represented as labeled or unlabeled nodes of a graph. Propositions may also have nodes associated with them. Finally, propositions in a semantic net are not assumed to be asserted (even though some researchers treat all nodes as implicitly asserted).

The proposition-based semantic network notation of Schubert (1974) is especially well suited for including pragmatic and semantic information as part of the meaning representation of individual word concepts. These meaning representations are networks based on propositions that consist of an n-ary predicate with a finite number of arguments. Terms used in the network to represent a given word concept can also be represented by semantic networks. Thus there is no insistence that a given set of "primitives" form the basis for the meaning of a word.[1]

The next section illustrates the use of semantic networks to represent the meanings of word concepts. Subsequent sections sketch methods that involve the computational use of these meaning representations in parsing and interpreting natural language text.

II. MEANING REPRESENTATIONS FOR WORD CONCEPTS

Cercone (1975) divides his lexicon into *open* class items and *closed* class items. Typically, closed classes have a strictly limited membership which cannot be increased by adding new formations or loanwords (which are words that have been incorporated by one language from another language). The significance of closed class items is best expressed by their grammatical function. In contrast, open classes have a large, readily increasing membership. New formations and loanwords are easily integrated.

Associated with open class category words are meaning representations: one for each sense of the word. The structure of a meaning representation is based on the semantic network notation developed by Schubert (1974). Pragmatic and semantic information are included in the meaning representation.

Figures 1 through 6 show networks that illustrate some of the main senses of the word drink, concentrating on action aspects. For illustrative purposes Figures 1, 3 and 6 are divided into a pragmatic section and a semantic section. The pragmatic section includes the template(s) that guides the parse of the utterance and two lists: the first list contains propositions that represent the implications that are likely to be needed for the comprehension of subsequent text; and the second list contains propositions representing critical implications that we expect to match in the surface structure. In Figure 1 this first list is (P3) and the second list is (P1,P2). The semantic section contains the network that represents the meaning of the word sense. Figures 2, 4, and 5 show various nominal senses of the word "drink".

Notice that Figures 1, 3, and 6 all have the notion of <u>change in containment location</u> in common. This corresponds to a <u>general concept</u> that subsumes not only differing senses of "drink" but also other more specific concepts as well, like "eating" or "receiving an enema". This observation has led to the following consideration.

When creating the meaning representations (networks) for concepts it is desirable to avoid the duplication of propositions in storage. If we extract more general concepts from the specific concepts that they subsume (totally or in part), we can avoid duplication by associating the common propositions with the more general concept.

In a sense the work of both Schank (1972) and Wilks (1973) supports the contention that the meaning of a concept is best represented by predications at the highest level of generality that adequately explain the term's meaning. Thus we extract from "drinking" (and eating, etc.) the structure shown in Figure 7.

We might reasonably label the concept expressed by this structure "ingest". It is important to note, however, that while Schank and Wilks might conclude that "ingesting" is a primitive action, that I consider it a general concept. This applies to all primitive actions put forward of Schank and Wilks. Examination of Figure 7 shows clearly that ingesting is <u>not a primitive</u> action but one whose meaning is expressed in terms of causes, motion, time, and other concepts.

At this point the original representations for the various action senses of "drink", i.e., Figures 1, 3, and 6, can be replaced with simplified diagrams based on the general concept "ingest". Figure 8 shows the representation of "drink" expressed in Figure 1 redrawn in terms of the general concept "ingest". In similar fashion Figure 9 diagrams one meaning of "eating", again based on the general concept "ingest".

The key to making effective use of the meaning representation for comprehension centers on the propositions that contain arguments that we expect to match in the surface utterance. The lexical item for "drink" would contain, among other things, pointers to a list of propositions; these propositions contain the arguments that we expect to match with words in the text and are most frequently needed for comprehension. At times, however, other propositions may be required for comprehension. For example, the word sense illustrated in Figure 1 shows that we expect to find, in an utterance about drinking, an anim(x) and a liquid(y) propositions P1 and P2. But the question can be posed, "What is the effect of John's drinking". To answer this question would entail a further investigation of the other propositions in the network, especially the first list of implications. Although it is implicit in the semantic structure, we make explicit in the pragmatic structure the inference that "x - drink - y" necessarily implies that it causes y's location to be in x at some time after x initiates the drinking action. Of course, since this implication is common to all senses of "drink" (and eats, inhales, etc.) it is abstracted into the same general concept "ingest" as well, as shown in Figure 7.

The semantic structure for each word sense for "drinks" is represented as properties attached to the word sense. The main properties include ARGS, the argument list containing arguments used in the word sense; IMPLICS, a list of implications that accompany the word sense; the propositions P1, P2, etc. that relate the arguments and predicates that make up the network explicating the given word sense; and templates of the form

arg1 arg2 ... argi WORD argi+1 ... argn

The implications make the most commonly used inferences part of the meaning representation of word concept. The propositions, for example P1 and P4 are shown Figure 10. See Cercone (1975) for sample lexical entries, in particular the entry for "drink".

Many advantages accrue by representing meaning formulas in this way. First, unlike Wilks' (1973) meaning formulas, the representation is suggestive of the meaning of a word. I see no justification for (binary) lexical decomposition trees as meaning representations for words as such trees are neither suggestive of the type of processing required nor of the propositions they encode.

A second and major advantage is this. The meaning representation for a word is not required to be explicitly in terms of "primitives". Rather, each of the predicates in the propositions that form the network representing the meaning of the word can, in turn, be represented in an analogous manner. In particular the notion of a "cause" seems to me to be no more "primitive" than "drink". This method of representing word meanings enhances the representational schema for the purpose of comprehension since any amount of detail can be included in the meaning representations by adding propositions to the networks.

Third, inference mechanisms, heuristic processing algorithms, and superimposed knowledge-organizing schemas can be incorporated using this representation for word meanings as easily as in any other representation. Incomplete information in surface text can be inferred, when necessary, directly from the meaning representation, in some cases as a missing argument.

The use of this type of meaning representation for lexical items is further explained in the next two sections.

III. PARSING AND INTERPRETATION USING NETWORKS

Traditionally, the object of parsing sentences has been to output syntactic trees. These trees served as input to semantic routines charged with the generation of meaning structures. Winograd (1972) and Woods (1970) tried, with some degree of success, to integrate the two processes and use each process to guide the other process. Schank (1972) and Wilks (1973) have stressed that syntactic processing was secondary to meaning analysis and should be necessary only when the resolution of ambiguity by meaning analysis alone had failed. Utilizing network meaning representations the parsing phase is almost completely semantically oriented. One important by-product in the method to be described is the detection of the correct <u>sense</u> of nominals, modifiers and actions.

The parsing proceeds as follows. Words, in a clause that has been classified[2] are scanned from left to right in search of a suitable candidate for an action. Once found, the sentence is separated into ((FIRST PART) (ACTION CANDI-

DATE) (SECOND PART)). The action candidate contains, among other things, a list of possible action senses that this particular root form may have. These senses are ordered by a scheme, albeit a very superficial scheme, described in Cercone (1975). Associated with word senses are templates as described above. For example, the sense *GIVE1 of the root form "give" has a template "X GIVE Y Z" and an alternative (ALTERN) template "X GIVE Z TO Y" associated with it.

The template, e.g. "X GIVE Y Z", is used to guide the parsing. In this example X, Y and Z are variables representing the arguments of the predicate "give" that we expect to find in the surface utterance in the given order. More detailed information concerning the arguments is obtained by examining the network propositions, for the sense of "give" in question, that involve the arguments. Thus X, in this case, would represent an ANIMATE nominal capable of "giving".

This is very similar to what Shcank does when parsing in conceptual dependency theory. If the words in the surface utterance do not satisfy the constraints for arguments of the predicate being examined, it is due to one of four reasons. First, alternate syntactic constructions could exist. Second, a different sense of the action is "correct". Third, the particular action candidate in question is not the action of the clause. Finally, some other reason, like slang expressions might be the cause.

Whenever arguments fail to satisfy predicates, a search for alternative implication templates begins. The result of this search is shown quite clearly, in Figure 11 of Section IV for the ternary predicate "give". In that example "give" is used syntactically in two different forms to distinguish the indirect object, one with the preposition TO and one without. If this approach fails then the list of senses for the root form is further examined. If other senses of the action candidate exist, they are examined further to see if arguments of the action candidate in the surface utterance match variables in the template. This procedure is repeated until the correct sense of the action candidate is found or the list of senses is exhausted. If the sense list is exhausted, scanning continues in the surface clause for another suitable action candidate and the process is repeated.

Part of the process of matching arguments of predicates in surface text to variables in implication templates involves finding the correct sense of nominals and modifiers as well. The sentence "A drinker drinks many drinks" has as the second argument of the predicate "drinks" the word "drinks". Possible nominal senses for that "drinks" include an alcoholic beverage, a body of water (throw John into the drink), or a thirst quencher. Thus, if the first sense of a nominal fails as argument, all other senses must be examined before deciding not to accept it as argument. This reasoning applies with respect to modifiers in a similar but not identical fashion. For instance, a "yellow cake" is a type of cake much like a chocolate cake whereas a "yellow car" is something that is yellow and something that is a car. Using these methods, sentences such as "A drinker drinks many drinks" and "The pilot banked his plane near the river bank over the bank that he banks on for good banking service" present little difficulty.

Morphological analysis is important since only those forms that can authentically be considered as actions need be examined. In the example, "A drinker drinks many drinks" the word "drinker" is eliminated immediately as an action candidate due to morphological analysis. Thus, we are very quickly able to get a right choice for an action candidate.

The next section shows an example of parsing and the resulting semantic network constructed using meaning representations of the type described.

IV. SOME EXAMPLES

The following example is taken from Cercone (1975). Many other examples can be found there. The sample listing preceding Figure 11 gives the results of the parsing phase, clause by clause, under the heading +++ ASSOCIATED ACTION-ARGUMENT-VARIABLE TRIPLES +++.

```
# R NEW:MACLISP
# 12:23.49
= (RESTORE 'CHKPT)
=     NIL
= (UNDERSTAND)
=     READY
=
= JOHN GAVE JUDY THE
= RED BOOK.  THEN, JUDY GAVE
= THE BROWN BOOK TO MARY.
=
=      +++ ASSOCIATED  ACTION-ARGUMENT-VARIABLE   TRIPLES +++
=       ((*GIVE1 *BOOK1 Z)  (*GIVE1 *JUDY1 Y)  (*GIVE1 *JOHN1  X))
=
=      +++MODIFIERS+++
=       ((NM (ADJ CLASF  ((0 0)  (*RED1))))  Z)
=
=         +++ THE SEMANTIC NET +++
=        *ATOM*      *VALUE*      *PROPERTY*
=      PROP0001      *JOHN1      X
=      PROP0001      *JUDY1      Y
=      PROP0002      *BOOK1      PRED
=      PROP0002      INST0003      ARG
=      PROP0004      INST0003      ARG
=      PROP0004      *UNS0005      PRED
=      PROP0001      INST0003      Z
=      PROP0006      *RED1      PRED
=      PROP0006      INST0003      ARG
=      PROP0001      *GIVE1      PRED
=
=      +++ ASSOCIATED   ACTION-ARGUMENT-VARIABLE   TRIPLES +++
=       ((*GIVE1 *MARY1 Y)  (*GIVE1 *BOOK1 Z)  (*GIVE1 *JUDY1  X))
=
=      +++MODIFIERS+++
=       ((NM (ADJ CLASF ((0 0)  (*BROWN1)))) Z)
=
=         +++ THE SEMANTIC NET +++
=        *ATOM*      *VALUE*      *PROPERTY*
=      PROP0007      *JUDY1      X
=      PROP0008      *BOOK1      PRED
=      PROP0008      INST0009      ARG
=      PROP0010      INST0009      ARG
=      PROP0010      *UNS0011      PRED
=      PROP0007      INST0009      Z
=      PROP0007      *MARY1      Y
=      PROP0012      *BROWN1      PRED
=      PROP0012      INST0009      ARG
=      PROP0007      *GIVE1      PRED
=      (MTS)
```

V. CONCLUSIONS

The above sections outline what I believe to be the correct approach to representing the meaning content of word concepts. Hopefully the use of meaning representations such as these will simplify the problems inherent in representing the conceptual content of natural language utterances in terms of meaning structures. In particular, I see the following desirable features inherent in this approach.

(i) Interpretive directness

The meaning structures corresponding to natural language utterances are formed according to simple structural rules. Powerful heuristic criteria, based on the central role of verbs and on preferred semantic categories for the subjects and objects of verbs, guide each choice in the creation of meaning structures. Interpretation of utterances then takes on a "slot and filler" character, rather than requiring extensive trial and error search.

(ii) De-emphasis of syntax

In ordinary discourse it would be absurd not to accept "ungrammatical" constructions like dangling participles or fanciful locations such as metaphor. In the above approach a syntactic straightjacket is not imposed on admissible utterances. Therefore the abnormal is not excluded as it is in many linguistic systems.

(iii) Emphasis on events

A major part of our interpretative effort in understanding natural language is focused on events, i.e., time-dependent relationships. By contrast, "static" relationships in the world are relatively easy to understand. Therefore the search for fundamental semantic structures should concentrate on the representation of events. The use of meaning representations as described above facilitates this emphasis on events.

The handling of vagueness, events, the lexical meanings of complex concepts, and the problem of overall knowledge organization may raise additional problems when processing natural language with meaning representations such as the ones I have used. However, the meaning representations used in this paper can be viewed as an extension of several successful but superficially disparate schemata, such as Schank's (1972) conceptualizations or Winston's (1970) descriptions. This indicates that their use should prove of real value in the design of understanding systems.

Acknowledgements

Many thanks are due to Dr. Len Schubert; his ideas and comments are interwoven in this research. I am also indebted to Dr. J. R. Sampson and Dr. K. V. Wilson for their careful reading and suggestions.

REFERENCES

Anderson, J. and Bower, G. (1973). Human Associative Memory, V. H. Winston and Sons, Washington, D.C.

Cercone, N. (1975). PhD thesis. University of Alberta. Edmonton, Alberta, Canada.

Cercone, N. and Schubert, I. K. (1974). "Toward a State-Based Conceptual Representation", Department of Computing Science, TR-74-19, University of Alberta, Edmonton, Alberta.

Davidson, D. and Harman, G. (eds) (1972). Semantics of Natural Language, D. Reidel Publishing Company, Boston, Massachusetts.

Fillmore, C. J. (1968). "The Case for Case", in Universals in Linguistic Theory, E. Bach and R. Harms (eds), Holt, Rinehart, and Winston, New York, New York, pp. 1-88.

Guzman, A. (1971). "Analysis of Curved Line Drawings Using Context and Global Information", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, v 6, pp. 325-375.

Palme, J. (1971). "Making Computers Understand Natural Language", in Artificial Intelligence and Heuristic Programming, N. Findler and B. Meltzer (eds), American Elsevier Publishing Company, New York, New York, pp. 199-248.

Quillian, M. (1968). "Semantic Memory", in Semantic Information Processing, M. Minsky (ed), MIT Press, Cambridge, Massachusetts, pp. 227-270.

Quillian, M. (1969). "The Teachable Language Comprehender", CACM, v 12, n 8, pp. 459-475.

Quine, W., (1970). Word and Object, MIT Press, Cambridge, Massachusetts.

Rumelhart, D., Lindsay, P., and Norman, D. (1972). "A Process Model for Long Term Memory", in Organization of Memory, E. Tulving and W. Donaldson (eds), Academic Press, New York, New York, pp. 198-221.

Schank, R. (1972). "Conceptual Dependency: A Theory of Natural Language Understanding", Cognitive Psychology, v 3, pp. 552-631.

Schank, R. (1973). "Identification of Conceptualizations Underlying Natural Language", in Computer Models of Thought and Language, R. Schank and K. Colby (eds), W. H. Freeman and Company, San Francisco, California.

Schank, R., (1975). "The Role of Memory in Language Processing", Yale University Technical Report, New Haven, Conn.

Schank, R., Goldman, N., Rieger, C., and Riesbeck, C. (1973). "Margie: Memory, Analysis, Response Generation, and Inference on English", Third International Joint Conference on Artificial Intelligence, SRI, Menlo Park, California, pp. 255-261.

Schubert, L. (1974). "On the Expressive Adequacy of Semantic Networks", TR74-18, Department of Computing Science, University of Alberta, Edmonton, Alberta. See also Proc. IJCAI-4.

Simmons, R. F., and Bruce, B. C. (1971). "Some Relations Between Predicate Calculus and Semantic Net Representation of Discourse", Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp. 524-530.

Wilks, Y. (1973). "Understanding Without Proofs", Third International Joint Conference on Artificial Intelligence, SRI, Menlo Park, California, pp. 270-277.

Winograd, T. (1972). Understanding Natural Language, Academic Press, New York, New York.

Winston, P. (1970). "Learning Structural Descriptions from Examples", PhD Thesis, MIT, MAC-TR-76, Cambridge, Massachusetts.

Woods, W. (1970). "Transition Networks for Natural Language Analysis", CACM, v 13, n 10, pp. 591-606.

FOOTNOTES

[1] Notable systems currently in vogue that utilize "primitives" in this way include those of Wilks (1973) and Schank et al. (1973).

[2] Words in clauses are morphologically analyzed and, based on that analysis, they are classified to determine all of their possible syntactic functions in an utterance.

[3] In Winograd's (1972) work, "gives" is recognized as a transitive action that requires two objects: his classification is TRANS2.

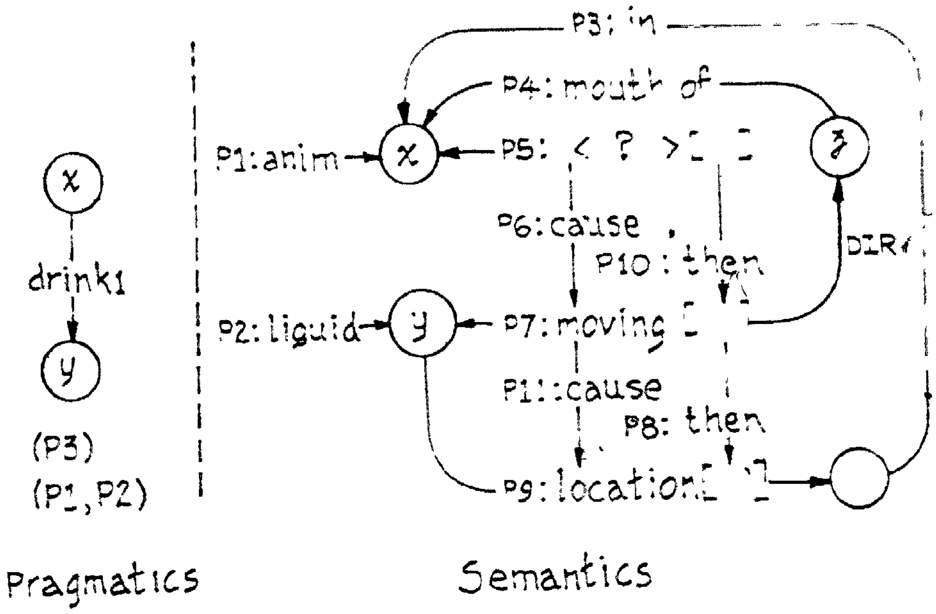


Fig 1. "(John) drinks (water)"
"(Mary) drinks (prune juice)"

Fig. 2. "(John is a) drinker"

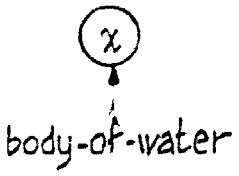


Fig. 4. "(Throw John in the) drink"

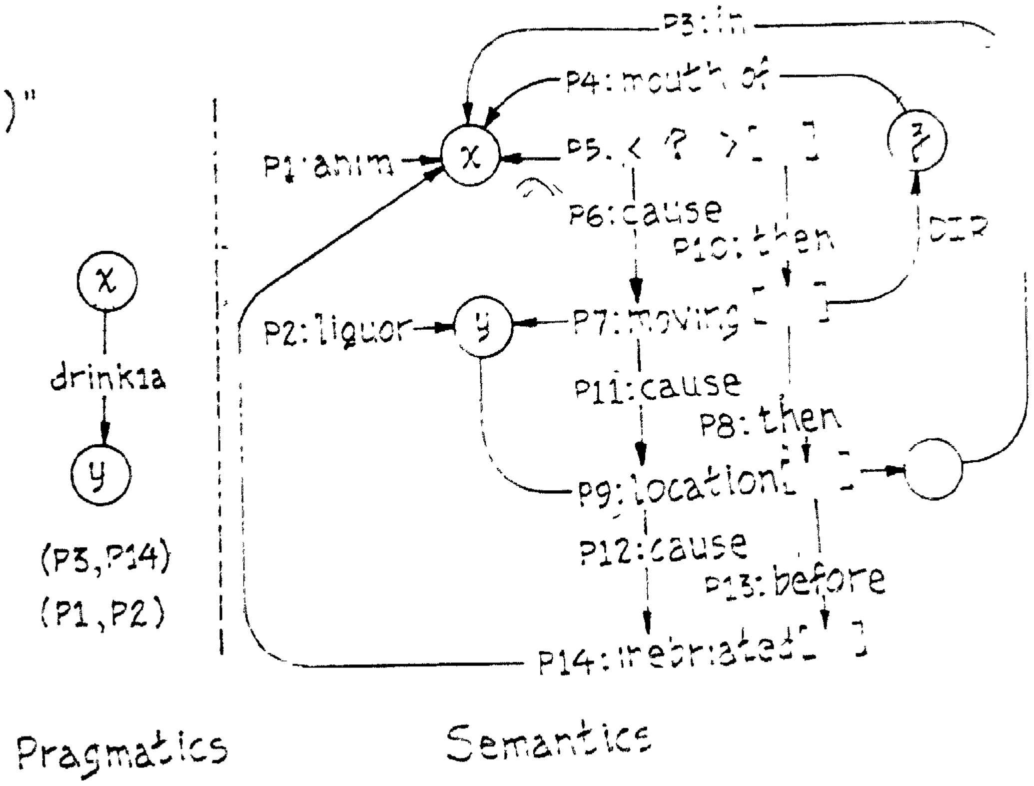


Fig. 3. "(John) drinks (whiskey)"
"(Mary has a) drinking (problem)"

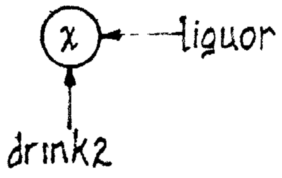


Fig. 5. "(John is drinking a) drink"

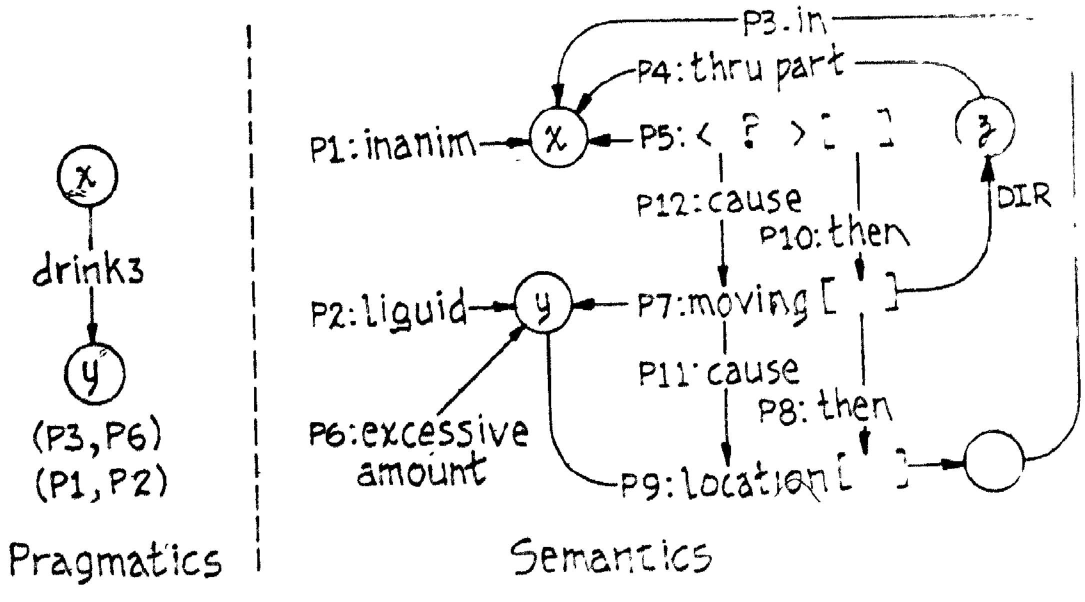


Fig. 6 "(My car) drinks (gasoline)"

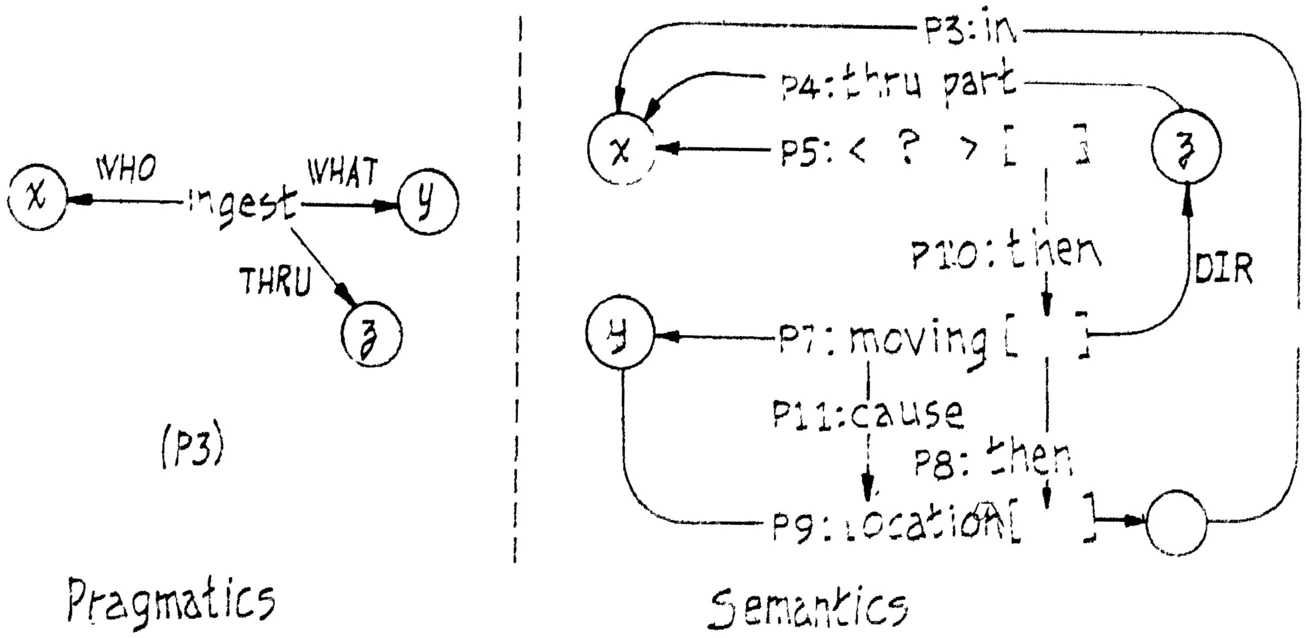


Fig. 7. "ingest"

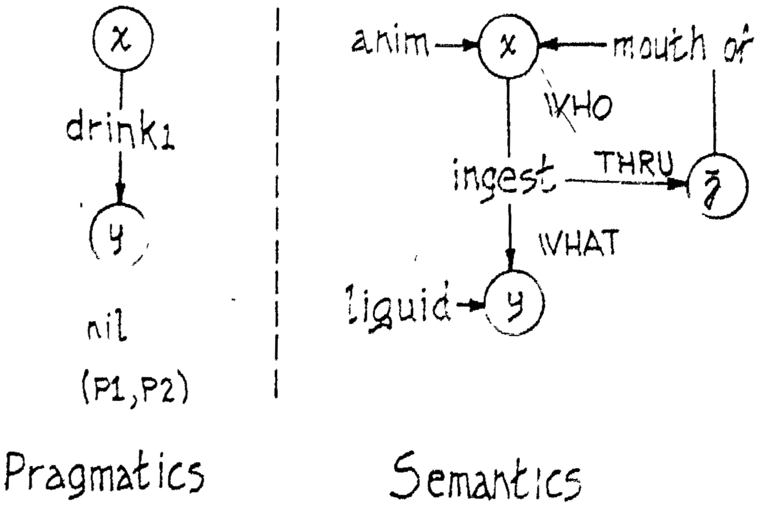


Fig. 8. "(John) drinks (water)"

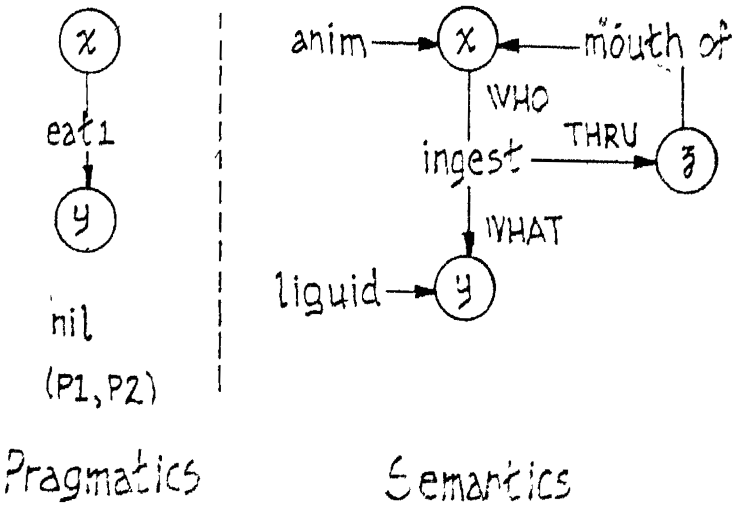


Fig. 9. "(John) eats (cake)"

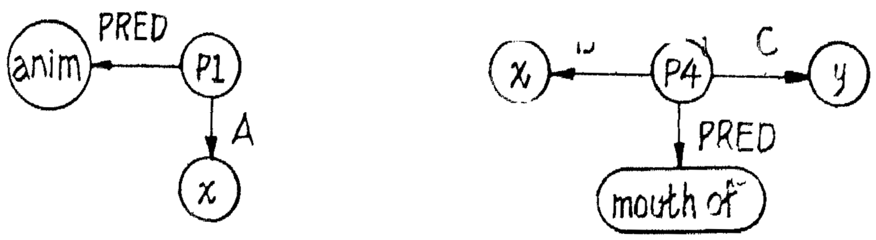
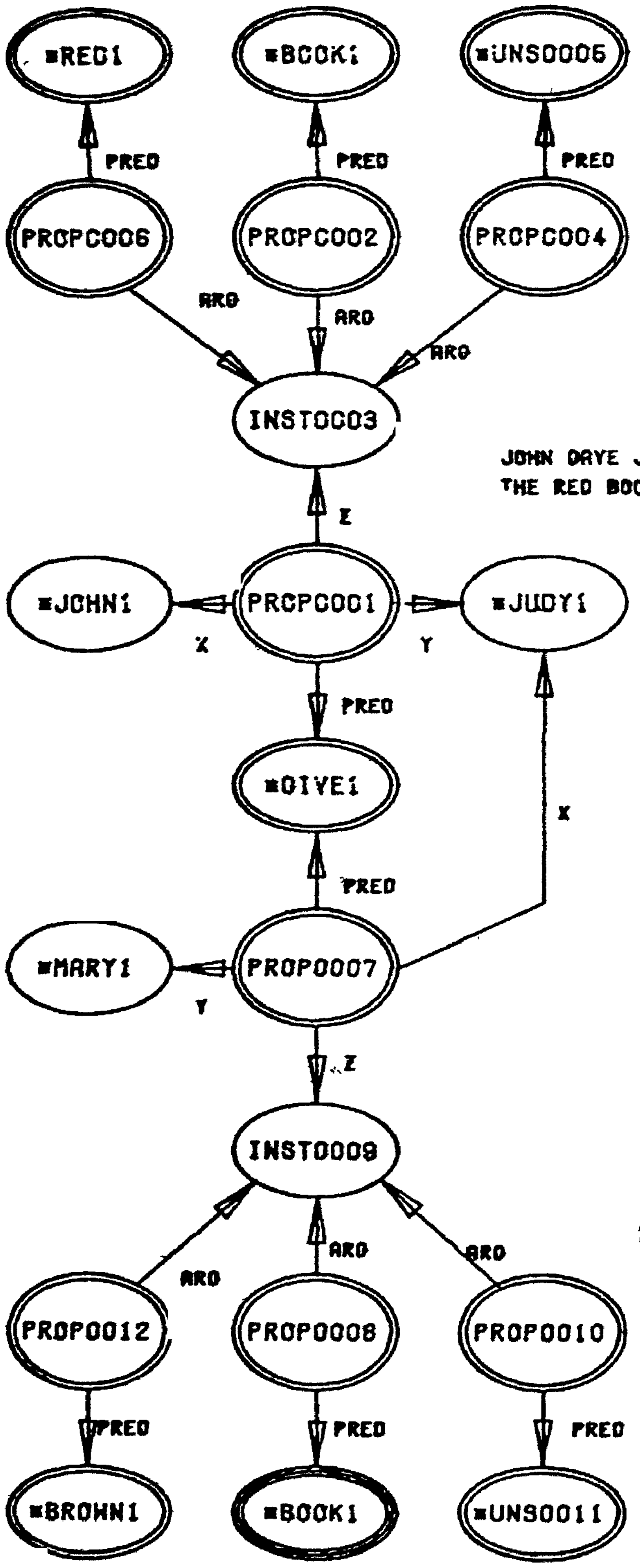


Fig. 10



JOHN GAVE JUDY
THE RED BOOK.

THEN, JUDY GAVE THE
BROWN BOOK TO MARY.

NOTES:
1. PROP LINKS ARE NOT SHOWN.
2. DOUBLY-CIRCLED NODES ARE PREDICATIVE CONCEPT NODES AND PROPOSITION NODES.
3. TRIPLY-CIRCLED NODES SPECIFY ALREADY EXISTING NODES BUT AID PERSPICUITY.
4. ARGUMENTS OF MONADIC PREDICATES ARE LABELLED ARG.
5. ARGUMENTS OF NON-MONADIC PREDICATES ARE ORDERED AND LABELLED X, Y, ETC.

Figure 11. N-ary Predicate Network

APPENDIX

Semantic networks present special problems with respect to the use of logical connectives, quantifiers, descriptions, modalities, and certain other constructions. Schubert (1974) has proposed systematic solutions to these problems by extending the expressive power of more or less conventional semantic network notation. In this appendix only the elementary part of the formalism, namely only as much as is necessary to clarify any misconceptions than may arise from the figures used in this paper, is explained.

In semantic network notation, the distinction between labels designating storage locations and labels designating pointers to storage locations requires clarification. This distinction is used by Quillian (1968) to designate "type nodes" (unique storage locations) versus "token nodes". The notation can be made uniformly explicit as in Figure A.1. Here "part-of", which in some notations corresponds to a token node, designates a type node (as suggested by Winston, 1970). All encircled nodes correspond to storage locations and all arrows to addresses of storage locations. What formerly were token nodes are now called proposition nodes; they serve as graphical nuclei for propositions as a whole.

At times the explicit notation of Figure A.1 will clutter the diagram leading to a loss in readability. Therefore, when the meaning is clear, binary predicates will be represented as in Figure A.2 for visual effect with the understanding that the use of explicit propositions underlie the structure.

In Figure A.1, A, B, and REL are mere distinguishing marks. They are analogous to parenthesis or commas in the Predicate Calculus in that they serve to relate denoting terms syntactically; they are non-denotative themselves. Whenever possible they will be chosen to be meaningful, i.e. to enhance readability and be suggestive, but they could be chosen as numeric labels as well.

One advantage of the explicit notation of Figure A.1 is that it works for n-ary (n>2) predicates. The sentence "John gives the book to Mary" involves "gives" as a three place predicate.[4] It is diagrammed as in Figure A.[3] Figure A.3 is appealing because of the significance we can attach to labels - agent, object, and recipient. By no means is Figure A.3 a graphical analogue of "case-structured" grammars. Cases are not viewed as conceptually primitive binary relations as Fillmore (1968) and researchers influenced by him, notably Schank (1972), view them. In a case structured system the central node would denote a specific action or process with the property that it is a "giving" and involves John, the book, and Mary as agent, object, and recipient respectively. Case relations can be understood as complex nonprimitive terms derived from such causally and teleologically related sequences of states. The whole notion of a case derives from the syntactic and semantic similarities between the role played by the arguments of many predicates. Nevertheless the notion of an "agent" seems to depend in part on causal priority of a state of the supposed agent in the sequence of states under consideration, and in part on the extent to which purposive behaviour can be ascribed to the supposed agent in general, and in part to the extent to which the particular sequence of states which he initiated can be assumed to be intentional on his part. See Cercone and Schubert (1974) for a further discussion of cases.

One final notational point by way of introduction needs to be made. The "case" labels in Figure A.3 are to be regarded as mere mnemonics, although indicative of more complex relations. To avoid confusion, predicate names will be designated in small letters and markers by capitals. Other conventions that are used include: solid loop for propositional nodes and existentially quantified concept nodes; broken loop for universally quantified concept nodes; solid lines to link the parts of a proposition to a proposition node; dotted lines for dependency links joining each existentially quantified node to all universally quantified nodes on which it depends; and broken lines for logical links.

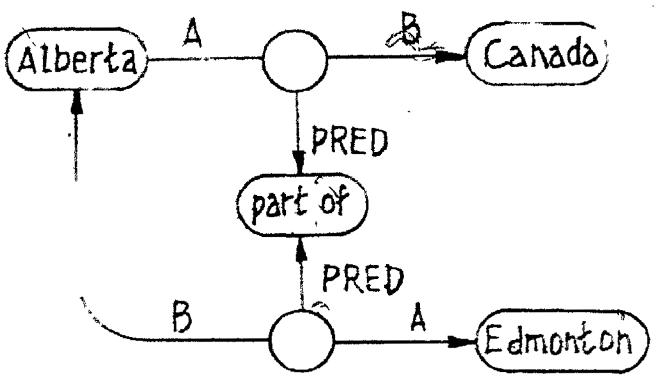


Fig. A.1 "Alberta is part of Canada.
Edmonton is part of Alberta."

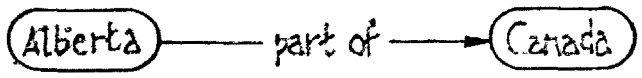


Fig. A.2 "Alberta is part of Canada

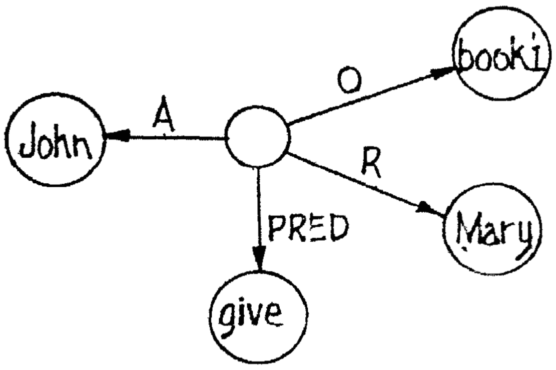
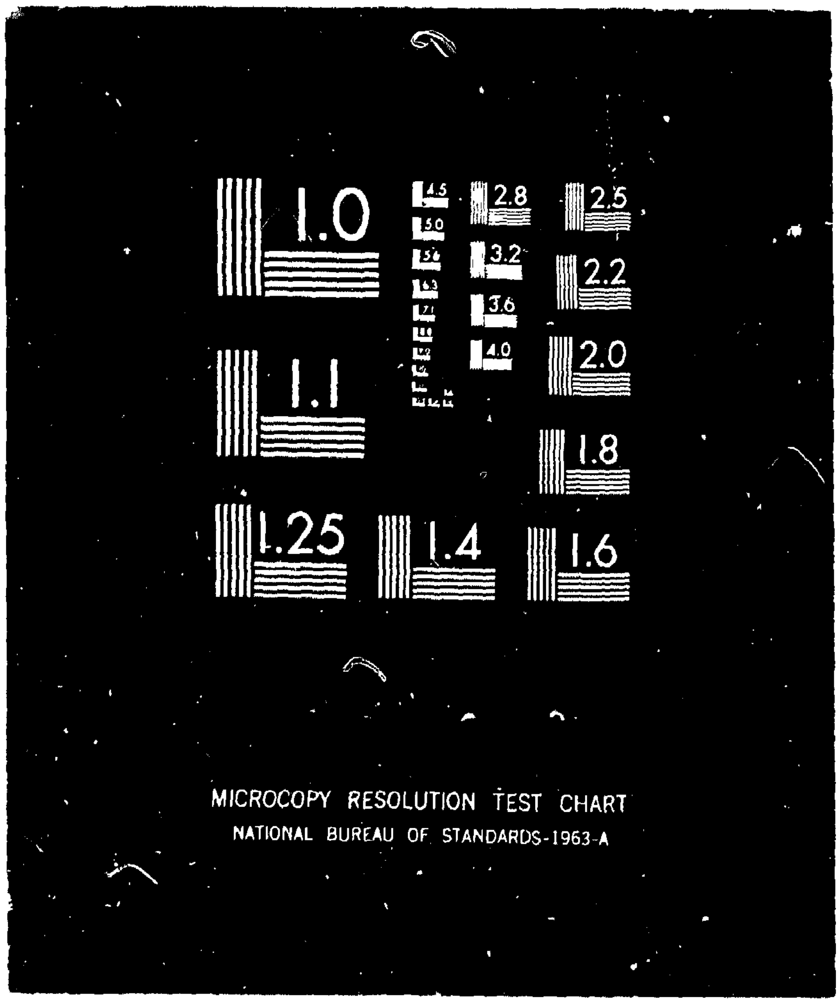


Fig. A.3. "John gives the book to Mary."

END


1.0
4.5
5.0
5.6
6.3
7.1
8.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A