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.

<u>Syntax, Semantics, and Pragmatics.</u> 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 Place case, identifies spaces which derive from the location of participants in its action, event, or state of affairs (or event/state). 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 Place objects.

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.

<u>SPS.</u> 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.

<u>SPS Rules and Locative Prepositions.</u> To see how SPS works in detail, and to

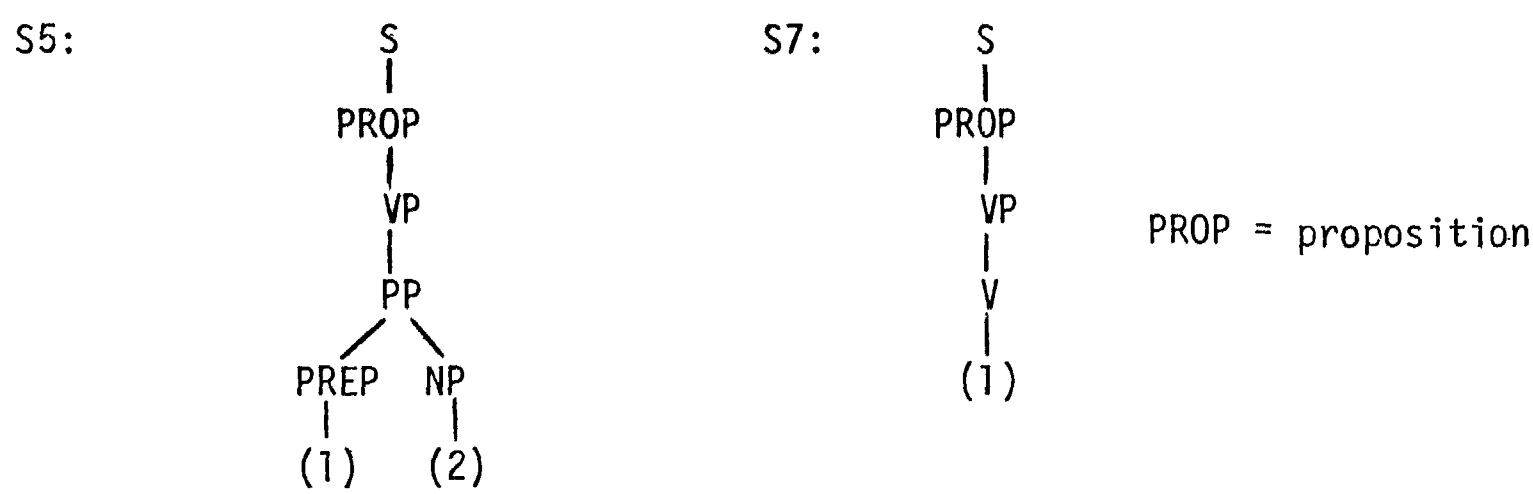
explain how it allows for locative prepositions we look at a typical rule:

```
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:



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

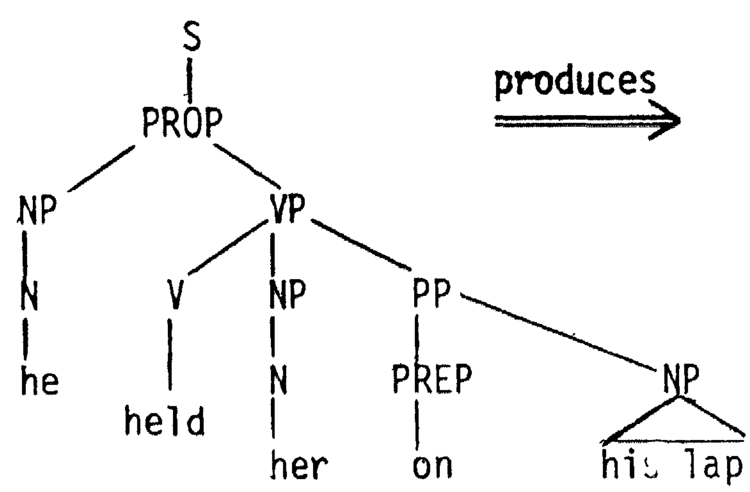
*With ambiguous entries, SPS tests each sense individually, therefore, any of the lexical references can be considered to have a unique meaning at any one time.

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.

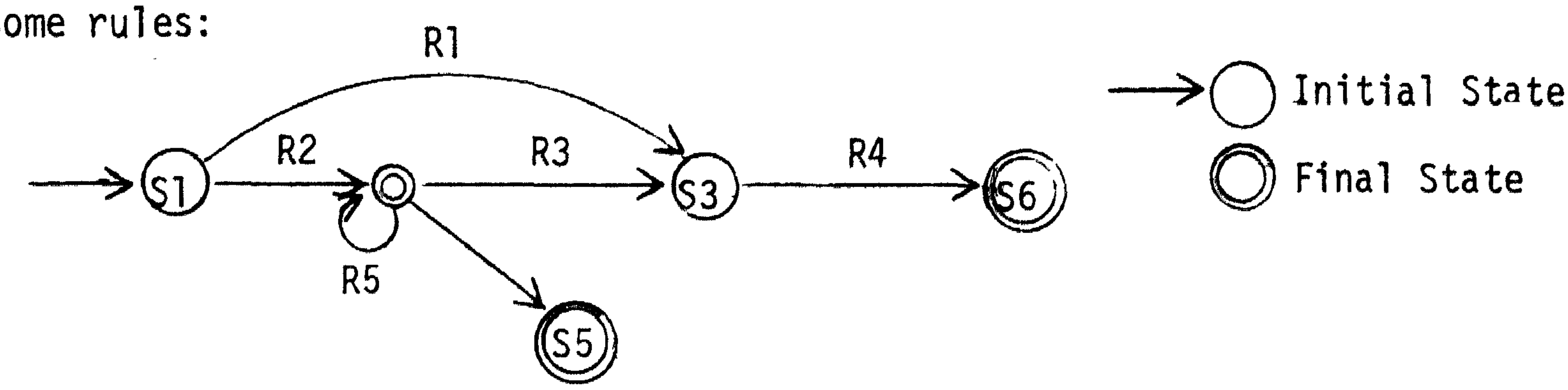
The two Place objects are formed by the use of a variable generation feature using the "!X("-number-")" form. References to the $BE event/state are also formed in this way. The other event/state is referenced through a register. SPS allows registers to hold variable names as well as feature sets. The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