American Journal of Computational Linguistics *Microfiche 32 : 21*

A General System for Semantic Analysis of English and its Use in Drawing Maps from Directions

Jerry R. Hobbs

Department of Computer Science
The City College of the
City University of New York
Convent Avenue at 140th Street
New York, New York 10031

ABSTRACT

We describe a semantic processor we are constructing which is intended to be of general applicability. It is designed around semantic operations which work on a structured data base of world knowledge to draw the appropriate inferences and to identify the same entities in different parts of the text. The semantic operations capitalize on the high degree of redundancy exhibited by all texts. Described are the operations for interpreting higher predicates, for detecting some intersentential relations, and in particular detail, for finding the antecedents of definite noun phrases. The processor is applied to the problem of drawing maps from directions. We describe a lattice-like representation intermediate between the linguistic representation of directions and the visual representation of maps.

OVERVIEW[1,2]

We are trying to construct a semantic processor of some

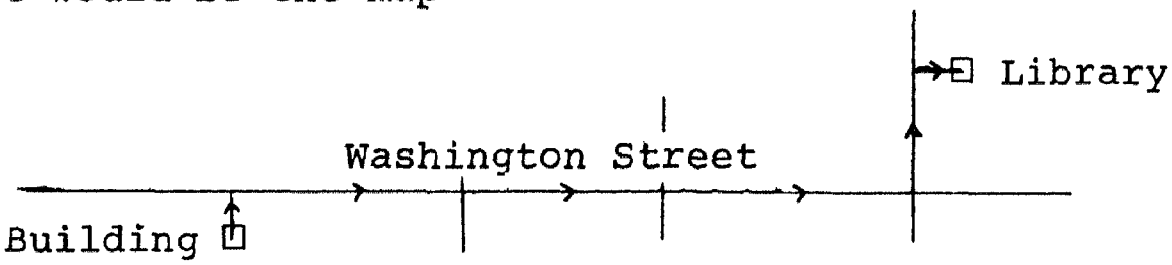
[1] This research was supported by the Research Foundation of the City University of New York under Faculty Grant No. 11233.
[2] The author would like to express his indebtedness to Harry Elam for many insights into the problems discussed here.

generality. We are using as our data base a set of facts involving spatial terms in English. To test the processor and to study the interfacing of semantic and task components, we are building a system which takes as input directions in English of how to get from one place to another and outputs a map, a map such as one might sketch for an unfamiliar region, hearing the directions over the phone.

A typical input might be the text

> "Upon leaving this building, turn right and follow Washington Street three blocks. Make a left. The library is on the right side of the street before the next corner." (1)

The output would be the map



To bypass syntactic problems, we are using as our input the output of the Linguistic String Project's transformational program (Grishman et al. 1973, Hobbs & Grishman), which is very close to a predicate-like notation. The semantic component is designed around general semantic operations which work on a structured data base of world knowledge to draw the appropriate inferences and to identify phrases in different parts of the text which refer to the same entity. The text, augmented and interrelated in this way, is then passed over to the task component, which makes arbitrary decisions when the map requires information not given by the directions and produces the map.

ORGANIZATION OF TEXT AND WORLD KNOWLEDGE

The two problems of semantic analysis are to find, out of a potentially enormous collection of inferences, the appropriate inferences, and to find them quickly. Our solution to the first is in our semantic operations described below. Our approach to the second problem is in the organization of the data base.

The data in the semantic component is of two sorts:

1. The Text: the information which is explicitly in the text. In the course of semantic processing this is augmented by information which is only implicit in the text. The text consists of the set of entities $X_1, X_2, \ldots$, explicitly and implicitly referred to in the text, and structures of the form $p(X_1, X_2)$ representing the statements made or implied about these entities, e.g.

$\text{walk}(X_1)$ = X_1 walks,

$\text{building}(X_2)$ = X_2 is a building,

$\text{door}(X_3, X_2)$ = X_3 is a door of X_2.

2. The World Knowledge or the Lexicon: the system's knowledge of words and the world. Words are the boundary between the Text and the Lexicon. A word is viewed as a key indexing a large body of facts (Holzman, 1971).

Associated with each word are a number of facts or inferences which can be drawn from the occurrence of $p(X_1, \ldots, X_n)$ in the Text. The facts are expressed in terms of p's set of parameters $Y_1, \ldots, Y_k$, and a set of other lexical variables $z_1, \ldots, z_m$, standing for entities whose existence is also implied. A fact consists of enabling conditions and conclusions. When $p(X_1, \ldots X_n)$ occurs in the Text and the semantic operations determine a

particular inference appropriate, its enabling conditions are checked. If they hold, the conclusions are instantiated by creating a copy of them in the Text with the lexical variables replaced by Text entities.

Clusters. One way to state the "frames" problem (Minsky 1974) is "How should the data base be organized to guide, confine, and make efficient the searches which the semantic operations require?" We approach this by dividing the sets of inferences into clusters according to topic and salience in the particular application. In the searches, the clusters are probed in order of their salience. In our application, the top-level cluster concerns the one-dimensional aspects of objects and actions. For example, the fact about a block that it is the distance between two intersections is in the cluster. If "around the block" is encountered, less salient clusters will have to be accessed to find information about the two-dimensional nature of blocks. The most important fact about an apartment building is that it is a building, to be represented by a square on the map. But if the directions take us inside the building, up the elevator, and along the hallway, the cluster of facts about the interiors of buildings must be accessed.

A self-organizing list (Knuth 1973) of the clusters is maintained--when a fact in a cluster is used, it becomes the top-level cluster--on the assumption that the text will continue to talk about the same thing.

The "Truth Status" of Inferences. In natural language, unlike mathematics, one is not always free to draw certain

inferences. We tag our inferences always, normally, or sometimes. These notions are defined operationally. An always inference is one we are always free to draw, such as that a street is a path through space. A normally inference is one we can draw if it is not explicitly contradicted elsewhere, such as that buildings have windows. A sometimes inference may be drawn if reinforced elsewhere, such as the fact used below that a building is by a street. This classification of inferences cuts across the cluster structure of the Lexicon.

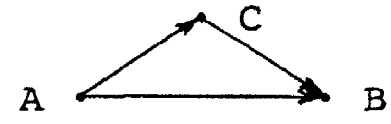
Lattices. A large number of statements in any natural language text, especially the texts this system analyzes, involve a transitive relation, or equivalently, say something about an underlying scale. For example, the word "walk" indicates a change of location along a path through space, or a distance scale; "turn" indicates a change along a scale of angular orientation.

In any particular type of text there are scales or transitive relations which are important enough to deserve a more economical representation than predicate notation. In this particular task, the important scales are a distance scale, a subscale of this indicating the path "you" will travel, and a scale representing angular orientation. This is the principal information used in constructing the map. For these scales we translate into a directed graph or lattice-like representation (Hobbs 1974).

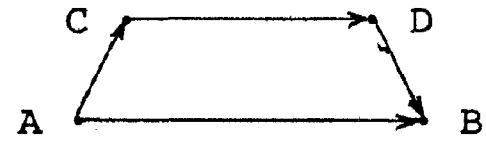
Some of the things which can be said about the structure of a scale are that some point is on the scale, that of two points on the scale one is closer to the positive end than the other,

and that a scale is a part of another scale. If a point B is closer to the positive end of the scale than point A, this fact is represented by

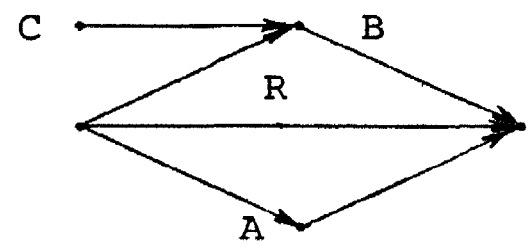
If point C lies in the interval from A to B the representation is



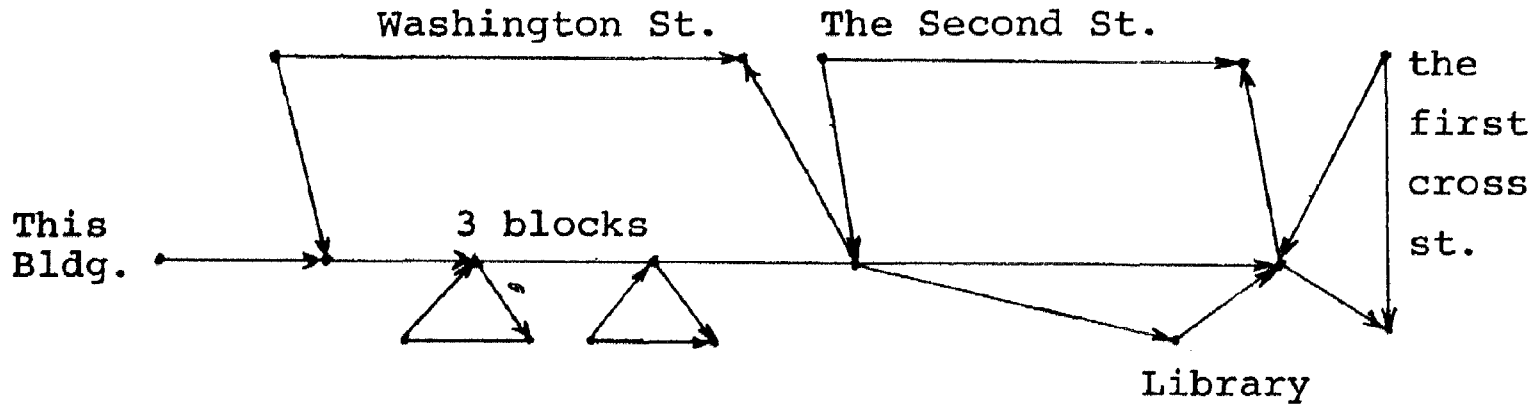
The diagram



means the scale from C to D is part of the scale from A to B. It is possible to represent incompleteness of information. For example, if it is known that points A and B both lie in a region R of a scale but their relative positions are not known and if it is known about C only that it precedes B this is represented by



The lattice for the distance scale for text (1) is as follows:



The lattices are intermediate between the linguistic representation of the directions and the visual representation of the maps. They are used at several points in the semantic and task

processes. They can be constructed for any transitive relation, and could be very useful, for example, in representing causal and enabling relations in a system translating descriptions of algorithms into flowcharts or programs.

SEMANTIC OPERATIONS

Basic Principle of Semantic Analysis. We believe the key to the first problem of semantic analysis, that of finding which inferences are appropriate, is Joos' Semantic Axiom Number One (Joos 1972), or what I will call the Principle of Knitting. Restated, this is, "The important facts in a text will be repeated, explicitly or implicity." That is, we capitalize on the very high degree of redundancy that characterizes all texts. Consider, for example, the simple sentence, "Walk out the door of this building." "Walk" implies motion from one place to another. "Out" implies motion from inside something to the outside. "Door" is something which permits motion from inside something to the outside or from the outside to the inside, or if closed, prevents this motion. "Building" is something whose purpose is for people to be in. Thus, all four content words of the sentence repeatedly key the same facts. Those inferences which should be drawn are those which are keyed by more than one element in the text.

This principle is used both formally and informally by the semantic operations. It is used formally in the interpretation of higher predicates and in finding antecedents. It is used more informally for deciding among competing plausible antecedents, resolving ambiguities, detecting intersentential relations, and knitting the text together in some minimal way. Here it is,

primarily the formal uses that will be described.

Interpretation of Higher Predicates. In "walk out", "walk slwoly", and "pleasant walk", the higher predicates "out", "slow" and "pleasant" all apply to "walk", but they narrow in on different aspects of walking. That is, each demands that a different inference be drawn from the statement that "X walks". "Out" and "slow" demand their arguments be motion from one place to another, forcing us to infer from "X walks" that "X goes from A to B". "Out" then adds information about the locations of A and B, while "slow" says something about the speed of this motion. "Pleasant", on the other hand, requires its argument to be an awareness, so we must infer from "X walks" that "X engages in a bodily activity he is aware of".

Stored in the Lexicon with each higher predicate is the inference which must be drawn from its argument and the information it adds to this inference. For example, "go(z_1,z_2,z_3)" must be inferred from the argument of "out". When the statement "out(walk(X_1))" is encountered in the Text, the higher predicate operation makes efforts to find a proof of "go(z_1,z_2,z_3)" from "walk(X_1)". The search for this inference is similar to the search procedure described below for finding antecedents. The facts in the resulting chain of inference are instantiated together with the information added by the higher predicate, and they are subsequently treated as though part of the explicit Text. It is usual for them to be useful in further processing, unless the modifier is simply gratuitous information.

Note that this operation allows considerable compression in

the number of senses that must be stored for each word. It allows us, for example, to define "slow" as something like "Find the most salient associated motion. Find the most specific Speed Scale for the object X of this motion. X's speed is on the lower end of this scale". This definition is adequate for such phrases as "walk slowly" (the most salient motion is the forward motion of the walking), "slow race" (the forward motion of the competitors), "slow horse" (its running at full speed, usually in a race), and "slow person". This last case is highly dependent on context, and could mean the person's physical acts in general, his mental processes, or the act he is engaged in at the moment.

This operation has a default feature. If a proof of the required inference can't be found, it is assumed anyway. This allows a text to be understood even if all the words aren't known. Suppose, for example, "veer right" is encountered, and the word "veer" isn't known, i.e. no inferences can be drawn from it. Since "right" requires a change in angular orientation as its argument, it is assumed this is what "veer" means. Only the information that the change is small is lost.

FIND ANTECEDENTS OF DEFINITE NOUN PHRASES

Entities referred to in a text may be arranged in a hierarchy according to their degree of specification:

1. proper names, including "you" and "I"
2. other noun phrases, including those with definite, indefinite, and demonstrative articles
3. third person pronouns
4. zeroed arguments and implied entities.

So far our work has concerned primarily definite noun phrases, but it is expected that many features of the definite noun phrase algorithm will carry over to other cases.

The definite noun phrase algorithm consists of four steps. First, "uniqueness conditions" are checked to determine whether an antecedent is required. If so, the Text and Lexicon are searched for plausible antecedents. Third, consistency checks are made on these. Finally if more than one plausible antecedent remains the Principle of Knitting is applied to decide between them.

Uniqueness Conditions. In the phrase "the end of the block", we know we must look back in the text for an explicitly or implicitly mentioned "block" (the search case), but we do not necessarily look for a previously mentioned "end" (the no-search case). Given a definite noun phrase the algorithm first tries to determine whether it belongs to the search or no-search case. This is done by checking two broad criteria. (These criteria were motivated by a large number of examples not only from sets of directions but also from technical and news articles.)

These criteria are checked by searching the Lexicon for certain features. However these searches are generally very shallow, in contrast to the potentially much deeper searches in the next step of the algorithm. Since by far the majority of definite noun phrases are in the no-search case, checking uniqueness conditions can result in great savings.

A caveat is in order. We state the criteria at a very high level of abstraction. We feel in fact that the algorithm can

work at that level of abstraction if the Lexicon is properly constructed. But how to construct a large Lexicon properly is a problem we have not yet tackled in detail. In any event, we give examples for each case, and the examples themselves form a reasonably exhaustive classification.

1. A definite entity is in the no-search case if it can be located precisely with respect to some framework. This includes the following conditions.

a. Objects which are located with respect to some identified point in space: "the building on the corner".

b. Plurals and mass nouns which are restricted to some identified region of space: "the trees in the park", "the water in the swimming pool". Here "the" indicates all such objects or substance.

c. Points and intervals in time which are fixed with respect to some identified event: "the minute you arrive", "the hour since you left".

d. Events in which at least some of the participants are identified and which can be recognized as occurring at a specific time: "the ride you took through the park yesterday".

e. Points or intervals on more abstract scales: "the end of the block", "the size of the building". The end is a specific point on the distance scale defined by the block. The size of the building is a specific point on the general size scale for objects, i.e. the volume scale.

f. Superlatives, ordinals, and related terms: "the largest house on the block", "the second house on the block", "the only

house on the block". If the set of comparison is identified, the superlative or ordinal indicates the scale of comparison and the place on that scale of the entity it describes. This is a subcase of (e).

All of these conditions can be checked in one operation if the facts in the Lexicon are expressed in terms of suitably abstract operators relating entities to scales. We simply ask if the definite entity is on or part of a scale or at a point on or along an interval of a scale, where the scale can be identified. However this requires that we take very seriously my suggestion in Hobbs (1974) that the lexicon for the entire language be built, insofar as possible, along the lines of a spatial metaphor. We have not yet had to face these problems since our only scales are physical -- our "at" and "on" are the locative "at" and "on".

Also checking this criterion presupposes a very sophisticated syntactic and semantic analysis. For example, (d) assumes that the times of events mentioned in tenseless constructions can be recovered.

2. A definite entity is in the no-search case if it is the dominant entity of that description. This divides into two sub-criteria:

a. Those entities which are unique or dominant by virtue of the properties which describe them: "the sun", "the wind". If the properties $p_1(X), p_2(X), \ldots$, are known about the definite entity X, the definitions of $p_1, p_2, \ldots$, are probed for the fact that the entity does not normally occur in the plural. Included under this heading are proper names beginning with "the", like

"the Empire State Building", and appositives, like "the city of Boston".

b. Those entities which are unique by virtue of the properties of an entity with which they are grammatically related: "the door of the building", "the Hudson River valley". "The door of the building" is represented in the Text as "$X_1|door(X_1,X_2|building(X_2))$" i.e. "the X_1 such that X_1 is the door of X_2 which is a building". The uniqueness or dominance of X_1 is not a property of "door" but of "building". Stored with "building" is the fact that a building has in its front surface a main door which does not normally occur in the plural. "The door of the building" is interpreted as this dominant door.

If the uniqueness conditions succeed, a pointer is set from the dominant lexical variable to the corresponding entity. If subsequently the same definite noun phrase occurs, the uniqueness check will discover this pointer and correctly identify the antecedent. Thus, we can handle the example

> "Walk up to the door of the building. Go through the door of the building."

Here the uniqueness check gives us a shortcut around the next step in the algorithm.

The Search for Plausible Antecedents. To illustrate the search for an antecedent, consider

> "Walk out the door of this building. Turn right. Walk to the end of the block."

What block? From "block" we follow a back pointer to the fact stored with "street" that "streets consist of blocks", and from

"street" the fact with "building" that "Buildings are by streets" Since a building is mentioned, we assume it is "the block of the street the building is on". The facts in the chain of inference leading to this are instantiated. An entity is introduced into the text for the "street" and the Text is augmented by the statements that "the building is on the street" and "the block is part of the street". This information turns out to be required for the map. Note that the fact that a building is on a street is a sometimes fact and that we are free to draw it only because "the block" occurs.

To conduct the search of the Lexicon, ideally we would like to send out a pulse from the word "block" which travels faster over more salient paths, and look for the first entity which the pulse reaches. The saliency is simulated by the cluster structure described above. The parallel process of the spreading signal is simulated by interleafing deeper probes from salient clusters with shallower probes from less salient clusters. For example, if "streets consist of blocks" is a cluster 1 fact, then we might probe for a cluster 1 fact involving streets and a cluster 2 fact involving blocks at roughly the same time. After one plausible antecedent is found in this way, the search is continued for possible antecedents which are nearly as plausible. If after a time no plausible antecedents are found, the search is discontinued.

Searches for antecedents are conducted not only for entities but also for definite noun phrases that the nominalization transformations of the syntactic component have turned into statements

--e.g. "The walk was tiring". Here we look back for a statement whose predicate is "walk" or from which a statement involving "walk" can be inferred. There are cases in which the required inference is in fact a summary of an entire paragraph--e.g. "These actions surprised..."--although of course we cannot handle these cases.

Consistency. Each of the plausible antecedents is checked for consistency. Suppose X_1 is the definite entity which prompted the search and its properties are

$$p(X_1), q_1(X_1), \ldots, q_m(X_1)$$

and X_2 is the proposed antecedent with properties

$$p(X_2), r_1(X_2), \ldots, r_n(X_2)$$

We must cycle through the q's and the r's to ensure they are consistent properties. Of course, to prove two properties q(X) and r(X) inconsistent can be an indefinitely long process with no assurance of termination. One admittedly ad hoc way we get around this is by placing into a special cluster those facts we feel are likely to lead quickly to a contradiction. The second tool we use for deriving inconsistencies may turn out to be quite significant.

In the course of processing, the lattice described above is constructed for several predicates. They contain information which can be useful in deriving an inconsistency. Suppose we have a text in which "the block" occurs explicitly several times. Toward the end of it, we encounter

> "Turn right onto Adams Street. The library is at the end of the block".

The search algorithm looks first for explicit mentions of "block" and finds them. Yet none of these entities is the one we want. Intuitively, the reason we know this is our almost visual feeling that we are already beyond those points.

The lattice consistency check corresponds precisely to this feeling. If a definite entity X_1 is a point or interval in a lattice or at a point or along an interval, we ask if the proposed antecedent X_2 is or can be related to a portion of the lattice. If so, then since the lattice represents a transitive relation, we need only ask if there is a path in the lattice from X_2 to X_1. If there is, they cannot be the same entity.

Many cases which pass for applications of the supposed recency principle--"Pick the most recent plausible antecedent"--are in reality examples of this consistency check. The earlier plausible antecedent is rejected because of lattice considerations.

As the text is processed, the whole structure of the discourse is built up. When a definite noun phrase is encountered, this discourse structure is known and it is this knowledge that is used to determine the antecedent rather than the linear ordering of the words on the page.

Competition among Remaining Plausible Antecedents. Even after the consistency checks, several plausible antecedents may remain, forcing us to decide among them on less certain criteria. To do this, we appeal to the Principle of Knitting again and make the choice that will maximize the redundancy in the simplest possible way.

A probe is sent out from the definite entity and from each plausible antecedent. Each plausible antecedent is searched for properties it has in common with the definite entity. Common properties count most if they are already in the Text, and within the Lexicon, common properties count more if they are within more salient clusters or they result from shorter chains of inference.

Default. Like the higher predicate algorithm, the definite noun phrase algorithm has a default feature. If the uniqueness conditions fail and the search turns up no antecedent, we simply introduce a new entity. In fact, in the directions texts there are a disproportionately large number of default cases, for "the object" may simply be the object you will see when you reach that point in following the directions.

Other Anaphora. We have not yet implemented routines for handling other anaphora. However, we believe they are very similar to the definite noun phrase routine, with certain differences. For entities tagged with demonstrative articles, we do not check uniqueness conditions, and the search will be narrower since the antecedent must be an entity or statement actually occurring in the text. For pronouns also, no uniqueness conditions are checked. The search will turn up more consistent plausible antecedents, and a correspondingly greater burden will be placed on the competition routine.

INTERSENTENTIAL CONNECTIVES

We detect unstated inter-sentence connectives by matching two successive sentences S_1 S_2 with a small number of common

patterns. In the directions texts the patterns are usually few and simple. The most common are

1. S_1 asserts a change whose final state is asserted or presupposed by S_2.

2. S_1 asserts or presupposes a state which is the initial state of a change asserted by S_2.

(These are likely very common patterns in all narratives.) For example, in the text

"Walk out the door of this building. Turn right.
Walk to the end of the block".

pattern(1) joins the first two sentences, where the state is "You at X". Pattern(2) joins the last two sentences, where again the state is "You at X". Note moreover that the sentences are interlocked by a second application of the two patterns: The first sentence assumes an angular orientation which is the initial state of the change asserted in the second sentence. The final state of this change is assumed by the third sentence.

In addition to providing the discourse with structure, this operation is one of the principal means by which implied entities in one sentence, like X above, are identified with those in another.

When pattern (2) is applied, we delete the independent occurrence of the state in the Text, so that subsequently it exists only as one intermediate state in a larger event. Changes across time are handled in this way.

TASK PERFORMANCE COMPONENT

Arbitrary Decisions. The semantic operations are quite

general and can be used for any application. The augmented and interrelated Text is then handed over to the task performance component, which of course is specific to the application.

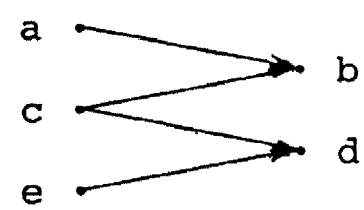
Our task component first makes arbitrary decisions required by the map but not given in the text. Both natural language directions and sketched maps allow information to be incomplete and imprecise, but in different ways. For example, in

"Turn right at the third street or the second stoplight".

we must decide whether to put the first stoplight at the first or second street.

The lattice representing the path "you" take must be complete in the sense that it is continuous, begins at the initial location, and ends at the desired goal, and that the relative locations of all points on the path are known. The lattice is complete if and only if there is a directed path passing through every point in the lattice at least once. If it is not complete, it is completed by supplying the fewest possible new links.

Geometrizing the Lattices. The second task operation is to convert the topological lattice representation into the geometric representation required by the maps. First we assign directions to all the points in the angular orientation lattice. In the simplest case we may have something like



where "a ⟶ b" means direction b results from a clockwise rotation of direction a. If no explicit directional information

is present, we simply assume a, c, and e are the same direction, and b and d are the same, and then assume the two directions are at right angles. Then in the distance lattice, contiguous or overlapping paths which share the same orientation are assumed to be parts of the same path and are mapped into a straight line. Information about names is accessed and assigned to the streets and buildings and the map is drawn.

Specific Systems with a General Semantic Component. We are aiming not so much at the construction of a general natural language processing system, which still seems reasonably far off but at an easier way of constructing specific systems. The case of syntax is instructive. It would be foolish for one who is building a natural language processing system to build his syntactic component from scratch. Large general grammars and parsers for them exist (e.g. Grishman et al. 1973, Sager & Grishman 1975). It is easier by several orders of magnitude to begin with a general grammar and specialize it, by weeding out the rules for constructions that don't occur in the texts one is dealing with, and by adding a few rules for constructions and constraints peculiar to one's application.

We are trying to make a similar facility available for the most common kinds of semantic processing. Specializing the general semantic component would consist of several relatively easy steps. First the Lexicon would be organized into a cluster structure appropriate to the task. At worst, this would mean specifying the necessary knowledge in a fairly simple format. If a very large Lexicon were available, this could mean no more

than designating for each fact the cluster it should appear in. Certain inferences could be made obligatory while others which are irrelevant to the task could be left out of the special Lexicon altogether. Second a Task Component would be built which would take, as ours does, the semantically processed Text, and use it to perform the task. We are demonstrating the usefulness of this approach in performing a task involving a visual representation. It is likely to be useful in other sorts of tasks also.

BIBLIOGRAPHY

Grishman, R., Sager, N., Raze, C., & Bookchin, B., "The Linguistic String Parser," Proc. NCC, AFIPS Press, Montvale, N.J. 1973.

Hobbs, J., "A Model for Natural Language Semantics, Part I: The Model," Yale Univ. Dept. Comp. Sci. Res. Rep. 36, Nov. 1974.

Hobbs, J., and Grishman, R., "The Automatic Transformational Analysis of English Sentences: An Implementation," Submitted to International Journal of Computer Mathematics.

Holzman, M., "Ellipsis in Discourse: Implications for Linguistic Analysis by Computer, The Child's Acquisition of Language, and Semantic Theory," Language and Speech (1971, 86-98.

Joos, M., "Semantic Axiom Number One," Language (1972) 257-265.

Knuth, D. The Art of Computer Programming, 3, Addison-Wesley, Reading, Mass., 1973.

Minsky, M., "A Framework for Representing Knowledge," MIT AI Memo 306, June 1974.

Sager, N., and Grishman, R., "The Restriction Language for Computer Grammars of Natural Language," CACM 18, 7 (7/75) 390-400.