COGNITIVE SCIENCE 13, 569–620 (1989)

# Marker Passing as a Weak Method for Text Inferencing

PETER NORVIG
*University of California, Berkeley*

The problem of deciding what is implied by a written text, of "reading between the lines" is the problem of text *inference*. To extract proper inferences from a text requires a great deal of general knowledge on the part of the reader. Past approaches have often used a "strong method" tuned to process a particular kind of knowledge structure (such as a script, or a plan). The alternative is a "weak method" which is applicable to a variety of knowledge structures. Just such a method is proposed here, one which recognizes six very general classes of inference. These classes are not dependent on individual knowledge structures, but instead rely on patterns of connectivity between concepts. Patterns are discovered, and inferences are suggested, by a process of marker passing between concepts.

## 1. THE PROBLEM OF INFERENCING

The reader of a text is faced with a formidable task: recognizing the individual words of the text, determining how they are structured into sentences, and deciding the explicit meaning of each sentence in the face of ambiguities. On top of that, the reader must also make *inferences* about the likely implicit meaning of each sentence, and the implicit connections between sentences. An *inference* is defined here to be any assertion which the reader comes to believe to be true as a result of reading the text, but which was not previously believed by the reader, and was not stated explicitly in the text.

It is important to distinguish between inference and deduction. Inferences need not follow logically or necessarily from the text; the reader can jump to conclusions that seem likely but are not 100% certain. Conversely, there can be assertions which necessarily follow from a text (are deducible from the text), but which are not inferred. For example, either Fermat's last theorem or its negation necessarily follows from any text on elementary

This work was supported in part by National Science Foundation grant IST-8208602 and by Defense Advanced Research Projects Agency contract N00039-84-C-0089.

Correspondence and requests for reprints should be sent to Peter Norvig, Computer Science Division, Department of Electrical Engineering and Computer Science, College of Engineering, University of California, Berkeley, CA 94720.

number theory, but so far no reader has been clever enough to infer which. Given this characterization, it is clear that *any* proposition whatsoever could be inferred from a given text. To avoid this unhappy situation, a more restricted characterization of inference must be given. This article will be concerned with *proper inferences,* inferences which satisfy the following three criteria:

1. They are *plausible;* not only is it possible for a reader to believe them, but it seems likely that the average reader would.
2. They are *relevant,* in that they serve to tie together concepts mentioned in the text.
3. They are *easy;* they seem to be made without conscious effort.

To understand a text, a reader must make the proper inferences, and avoid making improper inferences.

There has been much discussion in recent years on the best paradigm to approach this problem. Natural language has potentially unlimited expressiveness; it can be used to say almost anything. This suggests that a language-understanding system should be based on a general weak method, one which is capable of stringing together an arbitrary number of inferences to arrive at an interpretation. For example, one could adopt the metaphor of "understanding as deduction" and use a general purpose theorem prover. Such an approach would be near complete, but would be intractable for large problems.

On the other hand, it has long been recognized that (1) real-world knowledge plays an important role in language understanding, and (2) one should be concerned not with all possible texts, but rather with the restricted range of texts that actually are expressed. This suggests that it might be appropriate to incorporate some special purpose strong methods to recognize common patterns, and bring appropriate knowledge to bear. This approach would necessarily limit the number of inferences and thus the interpretations that could be made. The hope is that the interpretations could be limited to just those that are most likely to be expressed.

Many of the early (c. 1970s) text-understanding systems stressed this strong-method approach. There was a tendency to create new algorithms every time a new knowledge structure was proposed. For example, from the Schank school, one program, MARGIE, (Schank, 1973) handled single-sentence inferences. Another program, SAM, (Cullingford, 1978) was introduced to process stories referring to stereotypical situations or *scripts.* Yet another program, PAM, (Wilensky, 1978) dealt with *plan/goal* interactions. But in going from one program to the next a new algorithm always replaced the old one; it was not possible to incorporate previous results except by re-implementing them in the new formalism. It was not just that SAM "knew" something about scripts—SAM was built exclusively to process scripts, and only incidentally could it do other types of inferences. Similarly, PAM had

a control structure that was built to track character's goals and explain them in terms of plans and themes. Again, nongoal-related inferences were handled only incidentally. A notable attempt to combine knowledge about scripts, plans, and various other sources was made by Dyer (1982) in the BORIS program. In this system, different knowledge sources were considered, but only one at a time; there was no capability to combine evidence from different sources in order to make a final interpretation. BORIS was a production system with a large number of very specific rules (and some more general ones). The difficulty was in getting the rules to interact with each other, and in predicting the result of adding a new rule, or changing an existing one.

Because of these difficulties, the emphasis in text understanding has shifted in recent years towards weak-method approaches. Charniak and Goldman's (1988) WIMP, Hobbs, Stickel, Martin, and Edwards' (1988) TACITUS, Pollack and Pereira's (1988) Candide and Stallard's (1987) work have all taken this tack. In each case, there is an emphasis on solving various problems posed by the text, such as promotional reference, indefinite noun phrase resolution, quantifier scope, and discovering metonymic relations. The solution is constrained both by the text itself and by available world knowledge, which is coded in terms of probabilistic or nonmonotonic axioms. However, natural language texts are almost always ambiguous to some extent, so a unique solution to the constraint problem cannot be achieved. Instead, a least cost or most likely interpretation must somehow be computed, using heuristics that combine evidence from various sources while limiting the number of choices considered. Charniak and Goldman use probabilities throughout the process of interpretation, Hobbs et al. use a combination of cost functions and probabilities, Stallard uses a more ad hoc collection of heuristics, and Pollack and Pereira use strict composition, but with nonmonotonic interpretation rules. In each case, the interpretation rules are clearly specified, but what is not clear is the complexity of the resulting algorithm. Deduction is an intractable problem, and adding the uncertainty of doing abductive inference rather than deduction often makes things worse, not better. Hobbs et al., for example, point out that their algorithm performs very slowly when it is allowed to consider all possibilities, but when it is limited to searching for local interpretations first, it can be stuck with sub-optimal global interpretations. Realizing this problem, Pollack and Pereira describe an architecture that makes it easy to experiment with different evaluation schemes.

This article can be seen as a case study, an experimental point on the continuum from inadequate to intractable algorithms. It attempts to see how far one can go using a weak method with strict bounds on the amount of processing allowed. The processing is bounded in two ways. First, suggested inferences are found through a marker-passing mechanism which would operate in constant time on a highly parallel machine, and is efficient even in serial implementations, because of the new technique of *marker state*

propagation. Second, the suggested inferences are evaluated in a predetermined order which attempts to account for useful interactions of constraints, without having to consider all possible combinations of inferences exhaustively. Thus, like Hobbs et al., some global interpretations will be missed, but an attempt will be made to show that the trade-off is in some sense worth it.

The algorithm is intended to describe how proper inferences might be quickly computed, and is not intended to discover all types of inferences. Rather, a technique is presented for quickly finding a partial interpretation, which could then be used as input for further processing stages.

As an example of the scope of the problem, consider the following text, excerpted from a book of fairy tales (Anonymous, 1972, p. 80).

(1) In a poor fishing village built on an island not far from the coast of China, a young boy named Chang Lee lived with his widowed mother. Every day, little Chang bravely set off with his net, hoping to catch a few fish from the sea, which they could sell and have a little money to buy bread.

A reader of text (1) should be able to make proper inferences such as the following:

(2a) There is a sea which is used by the villagers for fishing, surrounds the island, and forms the coast of China.
(2b) Chang intends to trap fish in his net, which is a fishing net.
(2c) The word *which* in *which they could sell* refers to the fish.
(2d) The word *they* in *they could sell* refers to Chang and his mother.

The reader must also avoid making improper inferences. A representative set of *improper* inferences for (1) is listed below:

(3a) The villagers fish on a river in the middle of the island. The island is on a lake which is near the coast.
(3b) Chang will use the net, which is a butterfly net, as a deposit on a motor boat to go out fishing.
(3c) The word *which* in *which they could sell* refers to the sea.
(3d) The word *they* in *they could sell* refers to the fish.
(3e) The square root of 169 is 13.
(3f) Chang has a grandmother (who is perhaps deceased).
(3g) Chang lived with his mother.
(3h) Chang is wearing blue pants.

Most readers find it difficult to take seriously the inferences in (3a–d). They often have to go back to the text to see that (3a–d) are indeed possible at all; that they are not explicitly contradicted by the text. In fact, each of (3a–d) is consistent with everything stated in the text, they are just less plausible than the corresponding inferences in (2a–d).

In (3b) the motor boat seems irrelevant, and (3a) is a convoluted example that would fail on both the relevance and easiness criteria. While (3e) is

highly plausible, in fact is 100% certain, it is completely irrelevant. Although (3f) refers to a character in the story, it still is not a relevant inference because it does not tie together concepts from the text; it just adds peripheral information. One could go on from (3f) and infer that Chang's grandmother had a grandfather, and that he had a pancreas, and that this pancreas secreted insulin, and so on. In each case we have a highly plausible inference which is connected in some way to either the text itself or a previous inference. However, it is not enough for a fact to be inducible, or even deducible, from the text; (3e–f) are not proper inferences because the connection between them and the text fails to add anything to the interpretation of the story. Example (3g) is not considered an inference by definition here because it was stated explicitly in the text.

Text (1) was taken from a book where it was accompanied by some pictures. The illustrator presumably made inference (3h), because that is how Chang is depicted; (3h) will be referred to here as an *idiosyncratic inference.* People bring many such inferences to the interpretation of a text, but are still able to distinguish idiosyncratic inferences from proper ones. In order to draw a picture, it was necessary to choose some attire for Chang, but another choice, say, brown shorts, would have done as well. On the other hand, the picture in question also showed Chang with a fishing net; here a butterfly net could not have been substituted. The reader is aware that the author of the text intended for him to infer that the net is a fishing net, but did not have any intention one way or the other with regards to the color of Chang's clothes. Clark (1975) makes a similar distinction, speaking of authorized and unauthorized inferences.

There is an implicit contract between the author and reader wherein the author agrees to explicate enough of the situation so that the reader, by searching for proper inferences, can recover the rest of the information and make sense of the text. In a perfectly structured text there will always be easy, plausible connections between each pair of adjacent sentences. When these connections are missing, or when inferences prove to be incorrect, it is usually a signal that the writer is being humorous, ironic, mysterious, has a different view of the world, or is just being confusing. Indeed, much of what makes texts interesting is the intentional flaunting of this implicit contract.

Note that plausibility is a relative term. In the phrase *which they could sell* from (1), it is more plausible that *which* refers to the fish than to the sea, and that *they* refers to Chang and his mother rather than the fish. When faced with a choice of possible referents, it is possible to decide which is better using default assumptions like *the actors of selling events should be people,* and *the object of selling events should be products which are conventionally bought and sold.* When faced with no good referent, sometimes these assumptions have to be violated. Given sentence (4) below, the reader would be forced to infer that *which* refers to the sea and *they* refers to the

fish, even though this constitutes a very unusual event in the real world. Thus, it is easier to determine the better of two proposed interpretations of a sentence than it is to decide if any single interpretation is acceptable. In other words, there appears to be no threshold of acceptability in examples like these.

(4) The fish hoped to acquire a sea which they could sell.

It should be clear by this point that making proper inferences requires a great deal of knowledge. The reader must know the meanings of individual words, as well as the grammatical rules of the language. In addition, the reader must have specific world knowledge about the subject matter. For text (1), this would include knowledge of spatial relations (*in, on, near*); geography (*village, island, coast, sea, China*); familial relations (*boy, mother, widow*); commercial transactions (*buy, sell, have, money*), as well as other sources of knowledge. Collectively this will be called *common sense knowledge,* to distinguish it from expert and grammatical knowledge. Without this knowledge, one would be unable to decide among alternative interpretations of the text. For example, inference (2b) above, *Chang intends to trap fish in his net,* comes from a knowledge of nets, not from the structure of the input sentence. If the sentence had been (5) instead, the inference that Chang intends to trap fish in his dog would not be made.

(5) Chang set off with his dog, hoping to catch a few fish.

## 2. THE INFERENCING ALGORITHM

An inferencing algorithm in a program called FAUSTUS (Fact Activated Unified STory Understanding System) has been implemented. A preliminary version of this system was described in Norvig (1983a), and a complete account is given in Norvig (1986). The program is designed to handle a variety of texts, and to handle new subject matter by adding new knowledge rather than by changing the algorithm or adding new inference rules. Thus, the algorithm must work at a very general level, not by knowing a lot of specific inferencing tricks or rules. The goal of the algorithm can be stated succinctly as follows:

> Make those and only those inferences that serve as a direct connection tying together two concepts in the construal of the text, such that the connection is a simple one, and is the most plausible such connection.

"Construal" means the set of propositions derived from the parse of the text, augmented by the inferences that have been accepted. The marker-passing algorithm assures that all direct connections will be found, and that the corresponding inferences will be suggested. The suggestion evaluation mechanism then attempts to pick the most plausible subset of these suggested

inferences, such that no accepted inference contradicts another. This is not a complete theory of understanding, as there will be much that a reader can take from a text by making nondirect connections.

The system is most similar to Alterman's (1985) NEXUS. The difference is that NEXUS concentrated on the problem of enumerating a small class of connecting relations (class/subclass, sequence/subsequence, coordinate, antecedent, precedent, consequent, and sequel). FAUSTUS, on the other hand, concentrates on the connections themselves, and how chains of connections come together to suggest inferences. In simplest terms, the FAUSTUS algorithm is to read the input, translate it into a semantic network representation, and pass markers from each concept in the semantic translation to neighboring concepts in the network. When two markers reach the same concept, an inference may be suggested. The exact inference depends on the path taken by the two markers on their way to the collision point. The path (and hence the inference) is characterized solely in terms of the primitive links traversed, not by the domain-level concepts visited. This is what makes FAUSTUS' inference mechanism simple, yet extendible to an open-ended set of domain relations. After passing markers and suggesting inferences, the final step is to evaluate each suggested inference, resolve conflicts and accept each inference with no conflicting evidence. The algorithm can be broken into steps as follows:

**Step 0: Construct a knowledge base** defining general concepts like actions, locations, and physical objects, as well as specific concepts like bicycles and tax deductions. The same knowledge base is applied to all texts, whereas Steps 1–5 apply to an individual text.

**Step 1: Construct a semantic representation** of the next piece of the input text. Occasionally the resulting representation is vague, and FAUSTUS resolves vagueness in the input using two kinds of nonmarker-passing inferences. However, the parser cannot produce a genuinely ambiguous representation. The lack of interaction between parser and inference mechanism is a serious problem, but this article does not address the problem, nor the details of the parsing process.

**Step 2: Pass markers** from each concept in the semantic representation of the input text to adjacent nodes, following along links in the semantic net. Each marker points back to the marker that spawned it, so the marker *path* can always be traced from a given marker back to the original concept that initiated marker passing. Each marker also has a *marker state,* which controls which links it will spread across, and prevents markers from spreading indefinitely.

**Step 3: Suggest inferences** based on marker collisions. When two or more markers are passed to the same concept, a *marker collision* is said to have occurred. For each collision, look at the sequence of *primitive link types* along which markers were passed. This is called

the *path shape.* If it matches one of six predefined path shapes then an inference is suggested. Suggestions are kept on an *agenda,* rather than being evaluated immediately. Note that inferences are suggested solely on the basis of primitive link types, and are independent of the actual concepts mentioned in the text. The power of the algorithm comes from having the right set of suggestions (and of course the right predefined path shapes to implement the suggestions).

**Step 4: Evaluate potential inferences** on the agenda. The result can be either making the suggested inference, rejecting it, or deferring the decision by keeping the suggestion on an agenda. If there is explicit contradictory evidence, an inference can be rejected immediately. If there are multiple potential inferences competing with one another (as when there are several possible referents for a pronoun), and none of them is more plausible than the others, the decision is deferred. If there is no reason to reject or defer, then the suggested inference is accepted.

**Step 5: Repeat** Steps 1–4 for each piece of the text.

**Step 6: At the end of the text** there may be some suggested inferences remaining on the agenda. Evaluate them to see if they lead to any more inferences.

The knowledge base is modeled in the KODIAK representation language, a semantic net-based formalism with a fixed set of primitive links. A simplified version is presented for expository reasons; see Wilensky (1986) or Norvig (1986) for more details. KODIAK resembles KL-ONE (Brachman & Schmolze, 1985), and continues the renaissance of spreading activation approaches spurred by Fahlman (1979).

## 3. AN ANNOTATED EXAMPLE

This example shows the inferences that are generated by FAUSTUS in the course of processing text (1). FAUSTUS does not receive the text directly as input, instead it is passed a semantic representation of each input sentence; these are shown below as capitalized expressions in parenthesis preceded by "Rep:"

[1] Input: In a poor shing village built on an island
near the coast of China,

```
Rep: (VILLAGE (MOD = FISHING) (MOD = POOR)
              (LOCATION = A ISLAND) WHERE
              (BEING-AT (FIGURE = THE VILLAGE)
                        (GROUND = A ISLAND))
              (BEING-NEAR (FIGURE = THE VILLAGE)
                          (GROUND = A COAST (OF = CHINA)))
```

Inferring: a MOD of the VILLAGE is probably the PREDOMINANT-OCCUPATION
because the FISHING fits it best.
This is a RELATION-CONCRETION inference.

Inferring: the VILLAGE is a FISHING VILLAGE.
This is a CONCRETION inference.

The input says that the village is modified by the concept "fishing" in some unspecified manner. The program determines that fishing should be interpreted as the predominant occupation of the village. It is able to do this because of a collision between two marker paths that begin at "village" and end at "job." One path follows the links that says a village is a kind of polity, polities can have a predominant occupation, occupations are jobs of some kind, and fishing can be a job. The other half goes from the village, which has a mod relation, which is filled by the fishing activity, which is a kind of fishing, which can be a job. Associated with a path of this shape is the suggested inference that "mod" should be interpreted as the more specific relation, in this case "predominant-occupation." Once this assumption is made, the village can be further classified as a "fishing-village." Note that the knowledge base contains many other facts about fishing villages, such as the fact that they are usually near water. These facts are not considered, because there are no collisions involving them, and hence no suggestions. Both these inferences are called *concretion* inferences, because they take an abstract representation (like "mod") and interpret it as a more concrete one (like "predominant-occupation").

Inferring: a MOD of the VILLAGE is probably the AVERAGE-INCOME
because the POOR fits it best.
This is a RELATION-CONCRETION inference.

Rejecting: a MOD of the VILLAGE is probably the OVERALL-QUALITY
because another possibility, AVERAGE-INCOME, is more specific.

Determining how "poor" modifies "village" is difficult not only because the modifying relation is vague, but also because "poor" is ambiguous between "low in wealth" and "low in overall quality." The knowledge base says that people have incomes, polities have average incomes, and objects in general can have an overall quality level. Markers from the instance of "poor" in the input are therefore propagated to the concepts for people, polities, and things. Markers propagating from "village" reach polity and thing, and thus there are marker collisions at those two concepts. Each collision suggests an inference, but when FAUSTUS tries to evaluate the first of these two, it notices there is another inference competing with it, in the sense that accepting one of the two means rejecting the other. The evaluation rule for choosing between competing suggestions says to accept the most specific inference. In this case, the constrainer of average-income is

"polity," which is more specific than the constrainer of overall-quality, which is "thing." Thus, the average-income interpretation is accepted, and the overall-quality interpretation is rejected.

Notice that the possibility of interpreting "poor" as referring to income rather than average-income was never considered, because there was no person mentioned in the input, and thus no marker collision that would suggest that interpretation.

Another possible interpretation is that "poor" modifies "fishing" rather than "village." The whole phrase would then mean "a village where the fishing was not good." This interpretation cannot be considered by FAUSTUS because the input it gets—the output of the parser—has already specified the association between modifiers. The parser can return a representation that is ambiguous as to how something is modified, but it cannot return a representation that is ambiguous as to what modifies what. If FAUSTUS were better integrated with the parser, it could try both possibilities in turn. As it stands now, the parser must just guess at the proper parse, based on the subcategorization of the individual lexical items and on attachment preferences.

```
Inferring: the CHINA is viewed as a GEOGRAPHICAL-ENTITY.
    This is a VIEW-APPLICATION inference.

Inferring: a OF of the COAST is probably the LAND-BORDER
    because the CHINA fits it best.
    This is a RELATION-CONCRETION inference.
```

Here we see a *view application* inference. The knowledge base defines China as a country, which is a political entity. However, political entities cannot have coasts; only geographical entitites can. Part of the knowledge base is a general mapping, called a **view**, stating that political entities can be viewed as the geographical location they have jurisdiction over. So FAUSTUS infers that in this situation, China is being viewed as a geographical entity. After that is done, the ambiguous modifier "of" in the phrase "coast of China" can be resolved: Coasts have two components, a land-border and a water-border; China is known to be a land-mass, and thus can only fill one of those roles.

```
Inferring: there is a BODY-OF-WATER such that
    it is the LOCATION of the FISHING and
    it is the SURROUNDER of the ISLAND.
    This is a DOUBLE-ELABORATION inference.

Inferring: there is a BODY-OF-WATER such that
    it is the LOCATION of the FISHING and
    it is the WATER-BORDER of the COAST.
    This is a DOUBLE-ELABORATION inference.
```

Here are seen the first inferences that create something new, rather than just further specifying some ambiguous input. Because of this added complication, this inference will be explained in more depth. Figure 1 shows a small portion of the knowledge base, which was constructed before processing the text, as Step 0 of the algorithm. The concepts in the knowledge base are connected by links of a fixed set of primitives: D for the dominate (or a-kind-of) and I for the instance (or ISA) taxonomic relations, S for a slot in a frame (or equivalently a relation on a concept), and C for a type constraint on a relation. There are a few more primitive link types that will be explained later. Figure 2 shows how markers from fishing.1 and island.1, (two concepts in the representation of the first sentence) follow along paths in the semantic network and collide at the concept "body-of-water." Figure 3 shows the inference that was suggested by this collision, and adopted above.

Although no body of water was explicitly mentioned in the text, concepts that implicitly refer to a body of water were mentioned. In particular, there are three marker paths, starting at the fishing, the island, and the coast, that all collide at the concept body-of-water. Each of these is of the $\rightarrow$I$\rightarrow$S$\rightarrow$C$\rightarrow$ path shape. The three paths considered in pairs result in three collisions, and each collision suggests an inference. These are called *double-elaboration* inferences because they elaborate on two concepts at the same time by relating them to a third. Two of the suggested inferences are accepted, and lead to the results printed above. The third suggestion was that the body-of-water is the surrounder of the island and the water-border of the coast. This suggestion is now redundant because both of its components have already been adopted.

```
[2] Input: a young boy named Chang Lee lived with his widowed
           mother

Rep: (INHABITING (EXPERIENCER = A BOY (MOD = YOUNG-AGE)
                                      (NAMED = CHANG))
                 (WITH = A WIDOW A MOTHER (OF = THE BOY))
                 (LOCATION = THE VILLAGE))

Inferring: the EXPERIENCER of the INHABITING must be the
           INHABITER
   This is a RELATION-CLASSIFICATION inference.
```

This is an example of a nonmarker-passing inference. The input describes an inhabiting state with the experiencer being the boy. The definition of inhabiting in the knowledge base says that it is a kind of "being-at" state, with an inhabiter that plays the role of the experiencer of the state and the figure of the being-at. In other words, by definition any experiencer of an inhabiting must be an inhabiter. FAUSTUS recognizes this fact and prints the message. It is a nonmarker-passing inference because it was deduced

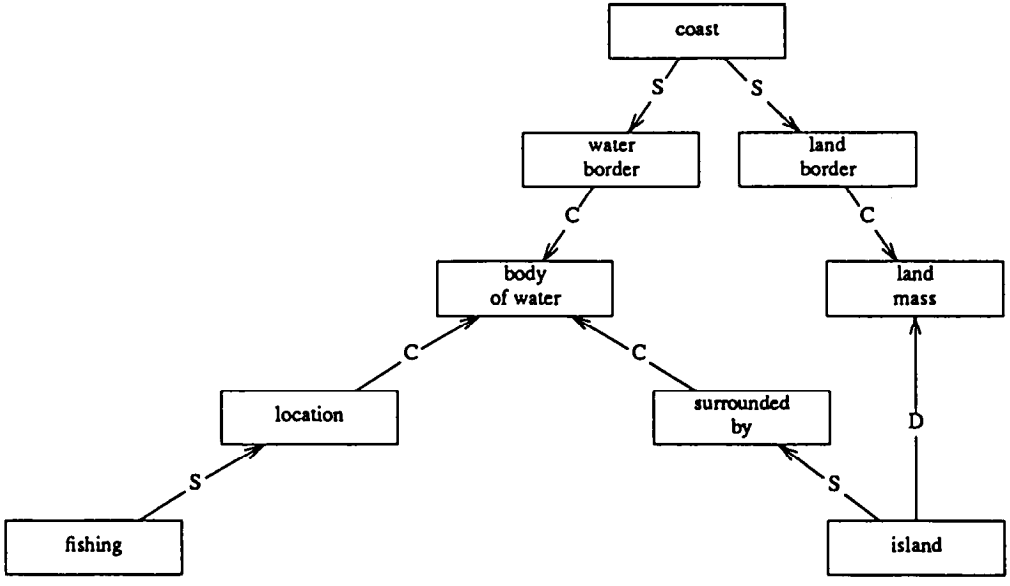


**Figure 1.** Part of the initial knowledge base network

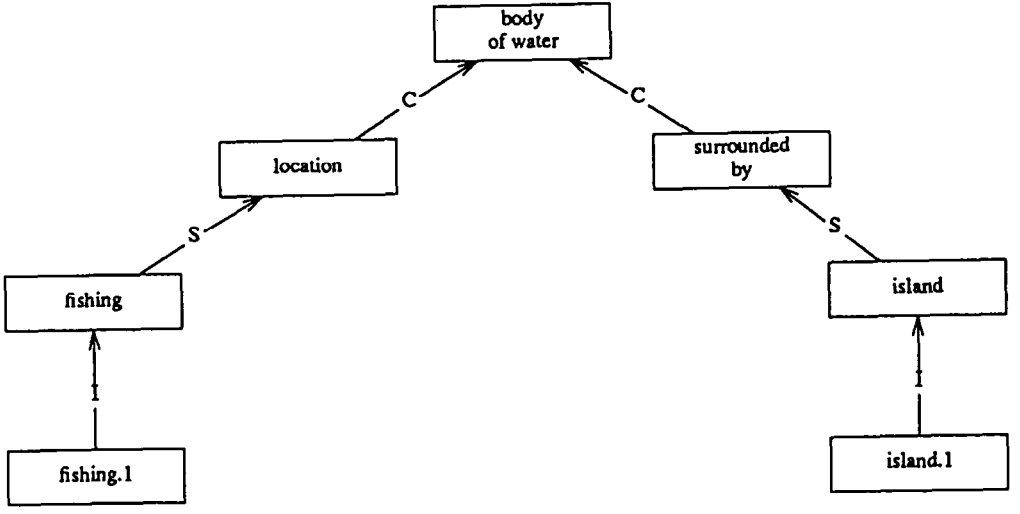


**Figure 2.** A marker collision

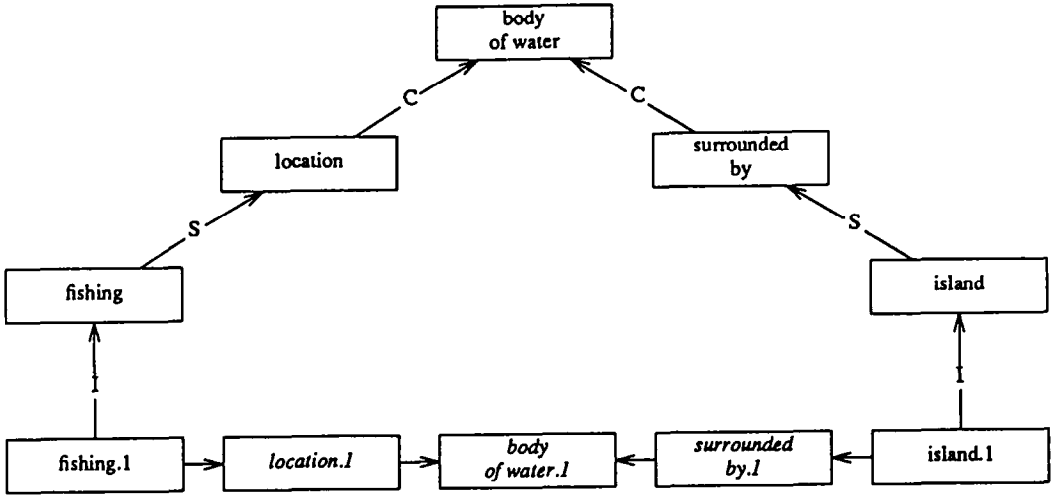


**Figure 3.** Inference suggested by marker collision

from the definition of inhabiting, not by a search procedure involving markers.

Inferring: a WITH of the INHABITING is probably the CO-INHABITER
because the MOTHER fits it best.
This is A RELATION-CONCRETION inference.

Inferring: a OF of the MOTHER is probably the OFFSPRING
because the BOY fits it best.
This is a RELATION-CONCRETION inference.

Inferring: the BOY must be a SON, because it is a OFFSPRING
This is a RELATION-CONSTRAINT inference.

Inferring: the INHABITING is a FAMILY-LIVING.
This is a CONCRETION inference.

Here there are two more cases of resolving ambiguous modifiers via concretion inferences. A "with" can mark an accompanier, an instrument, or a manner, but in this case there is a very specific type of accompanier, the co-inhabiter, that is compatible with "with." The mechanism for discovering this is a collision at "inhabiting" between a marker path originating at the instance of inhabiting, and the path originating at the instance of with. The path goes through the concept accompanier, and the suggested inference is that the "with" is actually an instance of accompanier. Once it is established that the inhabiting situation holds with the mother and son as participants, then it can be inferred that the inhabiting is an instance of family-living, a more specific situation known in the knowledge base.

[3] Input: Every day, little Chang set off with his net,

Rep: (TRAVELING (ACTOR = CHANG (MOD = SMALL-SIZE))
(WITH = A NET (OF = THE BOY)))

Inferring: the ACTOR of the TRAVELING must be the TRAVELER
This is a RELATION-CLASSIFICATION inference.

Inferring: a WITH of the TRAVELING is probably the ACCOMPANIER
because the NET fits it best.
This is a RELATION-CONCRETION inference.

In this case, there is no specific information on how "with a net" could modify an instance of traveling, so the default, the "accompanier case," is selected. Note that the phrase "every day" is ignored completely in the semantic translation. FAUSTUS does not have a sophisticated model of time or of habitual action.

[4] Input: hoping to catch a few fish from the sea,

Rep: (WANTING EXPERIENCER = THE BOY)
(WANTED = CATCHING (ACTOR = THE BOY)
(PATIENT = SOME FISH)
(SOURCE = THE SEA)))

Inferring: the EXPERIENCER of the WANTING must be the WANTER
This is a RELATION-CLASSIFICATION inference.

Inferring: the CATCHING must be a GOAL-SITUATION, because it is a WANTED
This is a RELATION-CONSTRAINT inference.

Inferring: the CATCHING is a CATCHING-FISH.
This is a CONCRETION inference.

Inferring: the SEA refers to the BODY-OF-WATER.
This is a REFERENCE inference.

Inferring: the NET is a INSTRUMENT of the CATCHING-FISH.
This is a SINGLE-ELABORATION inference.

Inferring: the NET must be a FISHING-NET,
because it is a CATCHING-FISH-INSTRUMENT
This is a RELATION-CONSTRAINT inference.

The first thing done here is to make the concretion inference that a catching action where the patient is some fish is actually an instance of catching-fish. This is detected because of a marker collision between one marker that starts at the fish and goes through fish to catching-fish and then up to catching, and another marker that starts at the fish, goes to the specific instance of catching, and up to the general concept catching. The action catching-fish is more specific than catching, and includes other information besides the fact that fish are caught. For instance, it is known that fish can be caught either in a net or on a line. Another connection is found by a marker collision at the concept trapping-device. One marker goes from the net up the hierarchy to trapping-device. The other marker starts at the catching, goes to the slot catching-instrument (the instrument of a catching action) and on to that slot's constrainer, trapping-device. Note that the two markers did not originate at the same time; such an inference serves to tie sentences together. After it is asserted that the net is the instrument of the catching, a nonmarker-passing, relation-constraint inference notices that the net can only satisfy the constraint on instruments of catching-fish if it is interpreted as a fishing net.

[5] Input: which they could sell

Rep: (SELLING GOAL (ACTOR = THEY) (PATIENT = WHICH))

Inferring: the ACTOR of the SELLING must be the SELLER
This is a RELATION-CLASSIFICATION inference.

Inferring: the PATIENT of the SELLING must be the THING-SOLD
This is a RELATION-CLASSIFICATION inference.

Inferring: 'THEY' refers to the FAMILY.
This is a REFERENCE inference.

Rejecting: 'THEY' refers to the FISH.
because the FISH is not a SENTIENT-AGENT.

FAUSTUS represents the word "they" as a group of unspecified nature. So markers are passed from the representation for "they" up the hierarchy to the concept group. Other marker paths that collide at group originate at the representation for the fish stated in input [4], and the family inferred in input [2]. These latter two paths collide with the first one at group, each suggesting a possible referent for "they." The reference is resolved because fish are not considered capable of performing a selling action. Note that if the program had not previously inferred the existence of the family (which was never mentioned explicitly), this inference could not be made.

Inferring: 'WHICH' refers to the FISH.
This is a REFERENCE inference.

The reference inference for the word "which" has many more possible referents. Ten other collisions (not shown) each suggest a referent, and the evaluation algorithm must choose between them. Several possibilities can be ruled out because they cannot play the role of thing-sold, a role that the word "which" is explicitly filling. Of the remaining possibilities, exactly one, the fish, was more recently mentioned than all the others. Thus, it is selected as the referent, and the others are rejected.

Inferring: there is a HAVING such that
it is a RESULT of the CATCHING and
it is a PRECONDITION of the SELLING
This is a DOUBLE-ELABORATION inference.

Here we have the introduction via a double elaboration inference of a new "having" state, wherein the family has posession of the fish. This state was inferred because it mediates between two other actions: It is the result of catching the fish, and is a precondition for selling them.

[6] Input: and have a little money

Rep: (HAVING GOAL (EXPERIENCER = THE GROUP)
(PATIENT = A MONEY))

Inferring: the EXPERIENCER of the HAVING must be the HAVER
This is a RELATION-CLASSIFICATION inference.

Inferring: the PATIENT of the HAVING must be the HAD
This is a RELATION-CLASSIFICATION inference.

Rejecting: the HAVING mentioned in [6] is a PRECONDITION of the SELLING.
because of a mismatch.

Inferring: the HAVING mentioned in [6] is a RESULT of the SELLING.
This is a SINGLE-ELABORATION inference.

Inferring: the MONEY is the PRICE of the SELLING.
This is a SINGLE-ELABORATION inference.

Here another instance of "having" is explicitly mentioned. FAUSTUS finds two single-elaboration connections between having and selling, but since the selling action above already has its precondition met, this one can only be the result.

[7] Input: to buy bread.

Rep: (BUYING GOAL (ACTOR = THE GROUP) (PATIENT = BREAD))

Inferring: the ACTOR of the BUYING must be the BUYER
This is a RELATION-CLASSIFICATION inference.

Inferring: the PATIENT of the BUYING must be the THING-BOUGHT
This is a RELATION-CLASSIFICATION inference.

Inferring: the BUYING is a PRECONDITION of the HAVING mentioned in [6].
This is a SINGLE-ELABORATION inference.

Inferring: the MONEY is the PRICE of the BUYING.
This is a SINGLE-ELABORATION inference.

Single-elaboration paths have found that the money can fill the price role in both the buying and selling. A more realistic interpretation might be that the money goes into the family cache, and is used a little at a time to buy bread, but FAUSTUS assumes that exactly the same money that was received from selling the fish is then used to buy bread.

## 4. EVALUATING THE ALGORITHM

One way to evaluate the FAUSTUS system is to see how well the inferences generated by the algorithm satisfy the criteria for proper inferences. Recall that the first implicit criterion was nonexplicitness. The algorithm avoids making inferences that were explicitly mentioned in the text by definition: No such inference would be suggested. The three other criteria were relevance, easiness, and plausibility.

First, the inferences are guaranteed to be relevant to at least two concepts in the model of the text, because inferences are only triggered when collisions occur. In other words, relevance is being *defined* by saying that an inference is relevant if and only if it is related (by marker paths) to two or more concepts in the model of the text.

The second criterion is easiness. This is guaranteed by designing the allowable marker paths so that only short paths will be allowed. The degree to which the length of a path corresponds to a human reader's intuitive judgement of relevance can be argued, but at least there is an explicit definition of easiness, which guarantees an upper bound on the amount of computing time required.

The final criterion is plausibility. Every inference that has been suggested because of a marker collision will be accepted, unless there is a specific evidence contradicting it. Contradictions are checked by a unification-like matching routine and can involve things like mismatching types on constraints or too many fillers for a relation. The matcher is discussed in the section on Step 4 below.

Thus, the FAUSTUS inferencing algorithm guarantees that all inferences will be easy, relevant, and plausible, at least according to the definitions of these terms laid out above. To evaluate the algorithm more thoroughly, one needs to know more details, as the next section will provide.

## 5. AN IN-DEPTH EXAMPLE

Here another example is presented. This time the program's trace level has been adjusted to print more verbose output, and to show the exact marker path shapes, collisions types, and inferences involved in processing text (6), a simple three-line text:

(6a) Bill had a bicycle.
(6b) John wanted it.
(6c) He gave it to him.

Understanding text (6) means making inferences like the following:

(7a) The word *it* in (6b) refers to the bicycle.
(7b) The word *he* in (6c) refers to Bill.
(7c) The word *it* in (6c) refers to the bicycle.
(7d) The word *him* in (6c) refers to John.
(7e) Bill having the bicycle is a precondition for giving it.
(7f) The giving results in John having the bicycle.
(7g) This new having satisfies John's goal of wanting the bicycle.
(7h) John wanting the bicycle was the reason for Bill giving it.

Different readers might have slightly different interpretations; the point of (7a–h) is that they give a likely interpretation and indicate the range and depth of the inferences that should be made. FAUSTUS' construal of the text is as follows:

Bill's Bicycle

[1] Bill had a bicycle.

Rep: (HAVING (EXPERIENCER = BILL) (PATIENT = A BICYCLE))

Inferring: the EXPERIENCER of the HAVING must be the HAVER
This is a RELATION-CLASSIFICATION inference.

Inferring: the PATIENT of the HAVING must be the HAD
This is a RELATION-CLASSIFICATION inference.

Int: (HAVING.1 (↑ HAVING) (haver = PERSON.1) (had = BICYCLE.1))

Above are seen two nomarker-passing inferences made during the process of turning the parser's representation into a KODIAK network. The resulting internal KODIAK representation is labeled with "Int:". There is a straightforward one-to-one correspondence between the "Rep:" and the "Int:"—the difference is that the latter refers to individual concepts, and has been made more specific by relation classification and relation constraint inferences.

At this point, markers are passed from the concepts in the internal representation to neighboring concepts. The blank space following the "Passing Markers and Suggesting Inferences:" below indicates that no inferences were suggested. When marker passing is complete, some summary statistics are printed. Then, since there are no suggestions to evaluate, the second input sentence is considered.

Passing Markers and Suggesting Inferences:

Concepts marked: 195 new, 195 total
Collisions: 91 new, 91 total
Suggested inferences: 0 new, 0 total
Accepted inferences: 0 new, 0 total

[2] John wanted it.

Rep: (WANT-TO-HAVE (EXPERIENCER = JOHN) (PATIENT = IT))

Inferring: the EXPERIENCER of the WANT-TO-HAVE must be the WANTER
This is a RELATION-CLASSIFICATION inference.

Inferring: the PATIENT of the WANT-TO-HAVE must be the THING-WANTED
This is a RELATION-CLASSIFICATION inference.

Int: (WANT-TO-HAVE.2 (↑ WANT-TO-HAVE)
(want-to-have = wanter = PERSON.2)
(thing-wanted = INANIMATE.2))

Passing Markers and Suggesting Inferences: 1 2

Each time there is a marker collision that leads to a suggested inference, FAUSTUS numbers the suggestion, prints the number, and places the suggestion on an agenda. We see above that inferences Number 1 and 2 have been suggested. The next step is to evaluate the suggestions:

```
Evaluating Suggestions:

Rejecting: the HAVING is a OUTCOME of the WANT-TO-HAVE.
   because of a mismatch.
   This is a SINGLE-ELABORATION inference.
   It is #1, due to the collision:
   WANT-TO-HAVE.2→elaboration→STATIVE←ref←HAVING.1

Inferring: 'IT' refers to the BICYCLE.
   This is a REFERENCE inference.
   It is #2, due to the collision:
   BICYCLE.1→ref→INANIMATE←ref←INANIMATE.2
```

Wanting to have something can lead to having it, and the marker-passing mechanism found a connection which suggests that Bill's having the bicycle is an outcome of John's wanting it. When it comes time to evaluate this suggestion, however, FAUSTUS checks it more carefully and decides there is a mismatch: The outcome should be a having with John as the haver, not Bill. Therefore, the suggestion is rejected.

The other suggestion was triggered by a collision at the concept INANIMATE. A bicycle is a kind of inanimate object, and the representation for 'it' is also an inanimate. Since 'it' needs a referent, this type of collision suggests that 'it' refers to the bicycle. When this suggestion is evaluated, there is no evidence to contradict it, and no competing referents for 'it,' so the suggestion is accepted.

Now the program prints statistics for the second sentence and moves on to the third:

```
Concepts marked: 25 new, 220 total
Collisions: 108 new, 199 total
Suggested inferences: 2 new, 2 total
Accepted inferences: 1 new, 1 total

[3] He gave it to him.

Rep: (GIVING (ACTOR = HE) (PATIENT = IT) (RECIPIENT = HIM))

Inferring: the ACTOR of the GIVING must be the GIVER
   This is a RELATION-CLASSIFICATION inference.

Inferring: 'HE' must be a SENTIENT-AGENT, because it is the GIVER
   This is a RELATION-CONSTRAINT inference.

Inferring: the PATIENT of the GIVING must be the GIVEN
   This is a RELATION-CLASSIFICATION inference.

Inferring: the RECIPIENT of the GIVING must be the GIVEE
   This is a RELATION-CLASSIFICATION inference.

Int: (GIVING.3 (↑ GIVING) (giver =MALE.3) (given = INANIMATE.3)
            (givee = MALE.3B))
```

In this sentence there are three relations: actor, patient, and recipient. Each becomes classified as a more specific relation, namely: giver, given, and givee. Again, these are called relation-classification inferences. Also here, is an instance of the other kind of nonmarker-passing inference: the relation-constraint inference. The idea is that anything that plays the role of a giver must be a sentient-agent: a person or some kind of agency or organization acting as a person. In the input, the giver is identified only as 'he'—a male animal—but not necessarily a person. FAUSTUS makes the inference that 'he' does in fact refer to a sentient-agent and therefore, a person. (Of course, there can be cases where the subject of a giving is a nonagent, such as *the music gave him a headache,* but such cases would be handled by a view interpretation that would map to a different lexical sense of "giving.")

Passing Markers and Suggesting Inferences: 3 4 5 6 7 8 9 10 11

Evaluating Suggestions:

Inferring: the 'IT' mentioned in [3] refers to the BICYCLE.
This is a REFERENCE inference.
It is a #5, due to the collision:
BICYCLE.1—ref—INANIMATE—ref—INANIMATE.3

Inferring: there is a HAVING such that
it is the RESULT of the GIVING and
the WANT-TO-HAVE is the SATISFIES of it.
This is a DOUBLE-ELABORATION inference.
It is #8, due to the collision:
GIVING.3—elaboration—HAVING—elaboration—WANT-TO-HAVE.1

Suggestion 5 is straightforward; since only one inanimate object, the bicycle, has been mentioned, it is the only candidate for the referent of 'it.' FAUSTUS therefore accepts the suggestion that they are coreferential. Suggestion 8 is a double-elaboration collision with origins at the want-to-have in Sentence 2 and the giving in Sentence 3. The suggestion is that there is a new 'having' situation wherein Bill has the bicycle, and this is the result of the giving and satisfies Bill's goal of wanting-to-have it.

Inferring: the WANT-TO-HAVE is the REASON of the GIVING.
This is a SINGLE-ELABORATION inference.
It is #9, due to the collision:
GIVING.3—elaboration—STATIVE—ref—WANT-TO-HAVE.2

Inferring: the HAVING mentioned in [1] is a PRECONDITION
of the GIVING.
This is a SINGLE-ELABORATION inference.
It is #10, due to the collision:
GIVING.3—elaboration—STATIVE—ref—HAVING.1
Rejecting: the HAVING mentioned in [1] is a RESULT of the GIVING.
because of a mismatch.

This is a SINGLE-ELABORATION inference.
It is #11, due to the collision:
GIVING.3→elaboration→STATIVE←ref←HAVING.1

Suggestions 9–11 are all single-elaboration inferences related to the giving action. These are because of marker collisions along the paths representing the following three facts: That somebody wanting something can be a reason for giving it to them; that having something is a necessary precondition of giving it; and that giving something results in someone else having it. The first two of these are accepted, thereby forming connections between the third sentence and the first two.

Inferring: 'HIM' refers to John.
This is a REFERENCE inference.
It is #3, due to the collision:
PERSON.2→ref→SENTIENT-AGENT←ref←MALE.3B

Rejecting: 'HIM' refers to Bill.
This is a REFERENCE inference.
It is #4, due to the collision:
PERSON.1→ref→SENTIENT-AGENT←ref←MALE.3B

Rejecting: 'HE' refers to John.
This is a REFERENCE inference.
It is #6, due to the collision:
PERSON.2→ref→SENTIENT-AGENT←ref←MALE.3

Inferring: 'HE' refers to Bill.
This is a REFERENCE inference.
It is #7, due to the collision:
PERSON.1→ref→SENTIENT-AGENT←ref←MALE.3

There are two reasons why a suggestion can be rejected. The first is a mismatch between two objects, as in Number 11 above. The other reason is that there are several mutually exclusive suggestions, of which only one can be accepted. It may be that none of the suggestions involves mismatches, but one still wants to be able to choose the "best" alternative. As examples of this, both John and Bill have been suggested as possible referents of the pronoun 'him' in Sentence 3. They are also both possible referents of the pronoun 'he.' At the time these suggestions were made, there was no reason to prefer one over the other. FAUSTUS tries to delay making a decision until more information is available, so it makes two passes over the suggestions, taking care of these suggestions in the second pass. That is the reason why Number 3 comes after Number 11 above. In this particular case, the wait was worthwhile, because Inferences 8 and 11 have added the information necessary to choose between the referents in each case. Inference 11 said that Bill having the bicycle was a precondition of the giving. But for this particular kind of precondition for giving, the knowledge base states

that the haver of the having must be the same as the giver of the giving. The haver is Bill, so when the suggestions are evaluated, the matcher quickly rules out John in Number 6 and accepts Bill in Number 7. Similarly, Inference 8 says that the result of the giving is a state where John is the haver of the bicycle, and that the haver and the givee are the same. Therefore, the matcher accepts John in Number 3 and rejects Bill in Number 4.

Note that it is a fact about English that the referents for 'he' and 'him' in Sentence 3 are necessarily distinct, and would necessarily be coreferential if 'himself' were used instead of 'him.' The current parser does not produce this information, but if it did, FAUSTUS could then make use of the information to limit possible referents.

After evaluating all the suggestions, the program prints the final statistics and stops.

```
Concepts marked: 59 new, 279 total
Collisions: 34 new, 233 total
Suggested inferences: 9 new, 11 total
Accepted inferences: 6 new, 7 total
```

## 6. STEP 0: REPRESENTING THE KNOWLEDGE BASE

To make these inferences requires some knowledge about bicycles, people, having, wanting, giving, and so forth. Representing this knowledge is Step 0 in the algorithm. The representation of these concepts is in no way dependent on the text of story (1). In fact, when it came time to make FAUSTUS process this text, the only information that had to be added was the definition of bicycle; all the other knowledge had already been defined for use in other texts.

A pictorial representation of a section of the knowledge network is shown in Figures 4–6. The diagrams can be paraphrased in English as follows. First, for Figure 4: In the lower half wanting is a kind of stative situation, which has a wanter and a wanted-state. One kind of wanting is wanting-to-have, where the wanted-state is constrained to be a particular kind of having, namely wanter-has-thing, which requires that the haver of the having must be the same person as the wanter of the wanting and the thing had in the having must be the same as the wanted-thing of the wanting-to-have. In the upper left, Bill and John are names, and there is a mapping from names to the people they refer to. The upper right shows that a bicycle is a kind of vehicle, and a vehicle is an artifact and also a functional-object whose purpose is travelling. Every functional-object has a purpose, which must be an action of some kind. Finally, in the lower right a certain kind of event is seen that can have a result, which is a kind of stative. In addition, there is something called a cause, and one particular type of cause can hold between events and the results of those events. There is also a

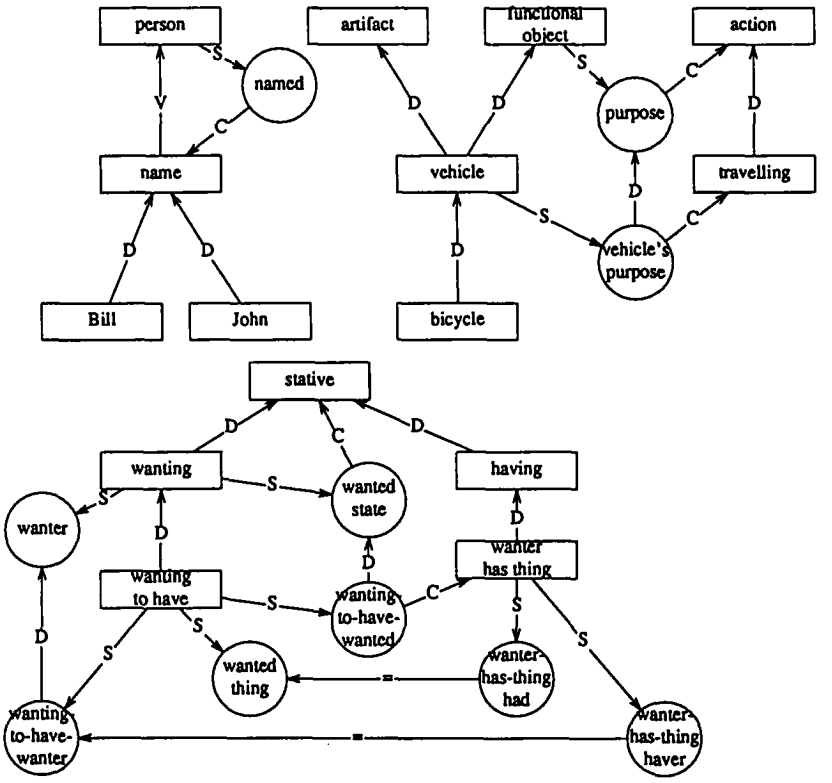


**Figure 4.** Domain knowledge: Person, bicycle, result, and wanting

particular before relation, called cause-before-effect which holds between the cause and the effect. Before has special semantics to the KODIAK interpreter in that it can only hold if the time of the before part is less than the time of the after part. The time of a state or event can be determined either by assertions that come explicitly in the input text, or by the default assumption that the order of presentation of the text is the same as the order of events in the world. In other words, when KODIAK is trying to determine if *A* occurred before *B*, it checks first to see if they participate in any explicit before or after relation, and if they do not, it goes on to look at the input times associated with *A* and *B*. Shown in Figures 4–6 are the eight primitive links among concepts, summarized in Table 1.

The relation between having and giving is shown in Figure 5. The concept having is defined to be a kind of strative situation, in particular it is a being, where the be-er is called the haver. The be-er is a participant in a situation, and thus must be a person. Besides having participants, situations can also have a patient, and the patient of a having is called the had, and must be a

TABLE 1
Primitive Links Among Concepts

| | Link Name | Description | Meaning |
|---|---|---|---|
| D | Dominate | X—D—Y | the class X is a subclass of Y |
| I | Instance | X—I—Y | X is an element of the class Y |
| V | View | X—V—Y | X's can be considered as Y's |
| S | Slot | X—S—R | every X has at least 1 R relation |
| F | Fill | R—F—Y | the value of this R relation is Y |
| C | Constraint | R—C—Y | every filler of a R relation is a Y |
| = | Equate | X—=—Y | X and Y are coreferential |
| ≠ | Differ | X—≠—Y | X and Y are not coreferential |

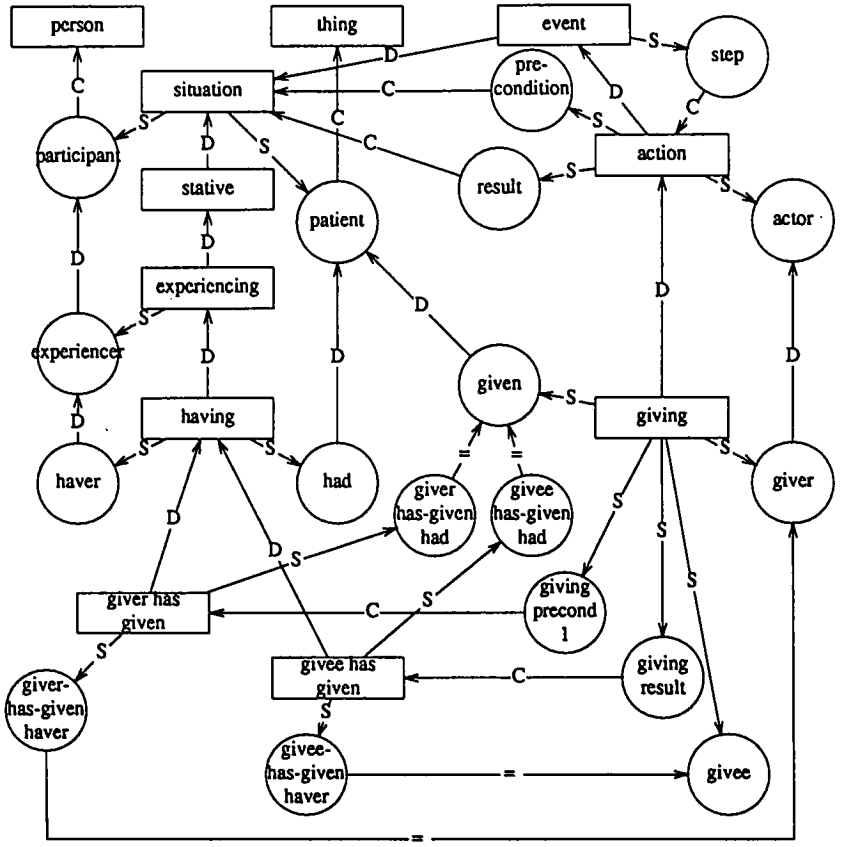


**Figure 5.** Domain knowledge: Having and giving

thing. The other kind of situation is an event, and one kind of event is an action. Actions have preconditions and a result. They also have an actor, which is a participant (although that link is not shown in the diagram). Giving is an action where the actor is called the giver, the patient is called the given, and there is another participant called the givee. (In the current world model, giving is also dominated by transferring, but that is not shown here.) Giving also has a precondition, namely giver-has-given and a result, givee-has-given, which are both kinds of having. However, the result of a particular instance of giving cannot be just any instance of a giver-has-given, it has to be one with the right object and the right recipient. We would not want the result of John giving a book to Mary to be that Bill has a pencil. Equate links are used to make the proper description. The equate links assert that the haver of the result of a giving must be the same as the givee of that giving, and the object given must be the same as the object had by the givee. Two similar equate links hold for the precondition.

Figure 6 describes two complex events related to giving, namely gift-giving and lending-functional-object. Gift-giving is defined as having two steps; buying-gift and giving-gift. The two steps are defined in more detail in the FAUSTUS knowledge base, but the detail is not shown here. The other event is lending-functional-object, which has three steps. First the lender gives out the object; then the lendee uses it for its intended purpose; then he returns it. One fact about this situation is depicted: The patient of the lending is the same as the object given, the object lent, and the object returned.

## 7. STEP 1: REPRESENTING THE INPUT TEXT

Now that the representation of some of the background knowledege necessary to understand text (1) has been seen, Step 1 of the algorithm can be considered: the representation of input text. The three sentences are represented in Figure 7. Not all of the instance links are shown; bicycle.1 should be an instance of bicycle, male.3b should be an instance of male, and so on. The notation having.1, for example, means that the event was mentioned in Sentence 1; unless mentioned otherwise it will be assumed that this occurred before events in Sentence 2.

### 7.1 NonMarker-Passing Inferences

Two classes of inferences happen immediately when the input text is processed, rather than as a result of the marker-passing algorithm. These inferences are called *relation-classification inferences* and *relation-constraint inferences.* Both are demonstrated in the following excerpt:

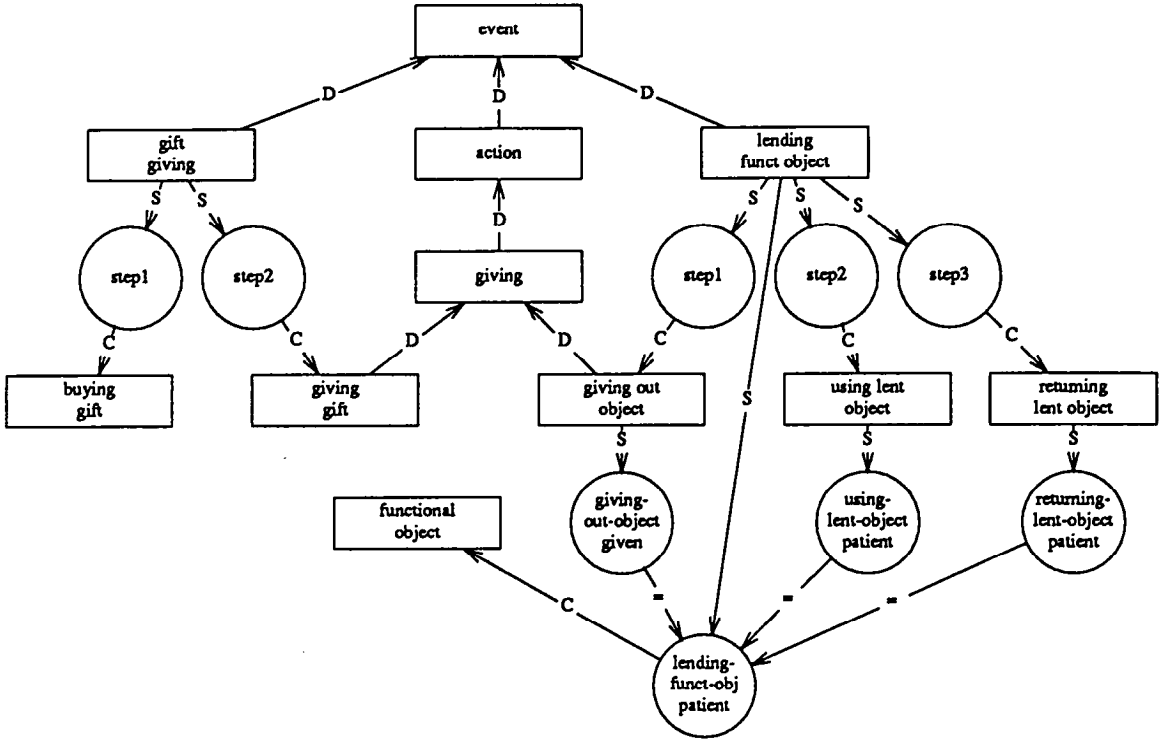


**Figure 6.** Domain knowledge: Gift giving and lending

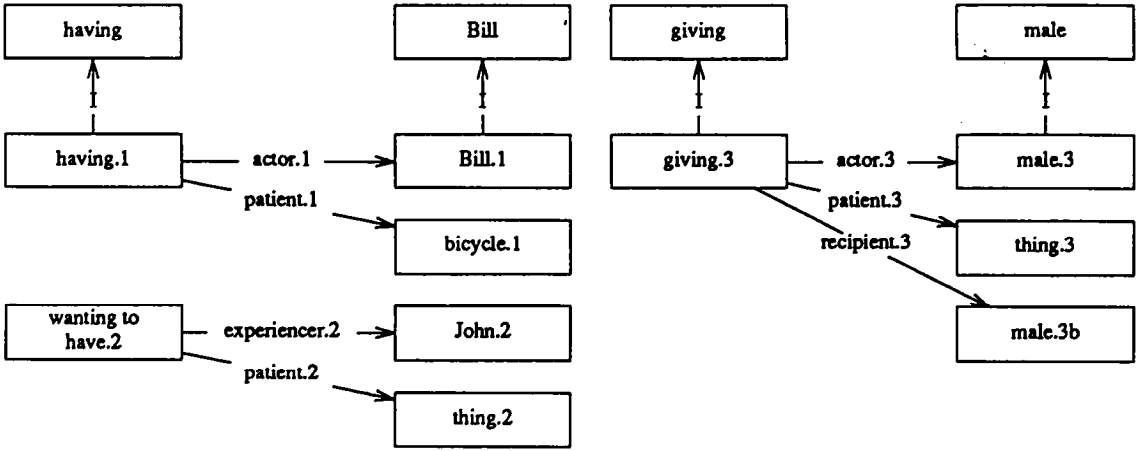


**Figure 7.** Representation of input text

[2] A girl started talking to him.

Rep: (TALKING (ACTOR = A GIRL) (PATIENT = HIM))

Inferring: the ACTOR of the TALKING must be the TALKER
This is a RELATION-CLASSIFICATION inference.

Inferring: the PATIENT of the TALKING must be the TALKEE
This is a RELATION-CLASSIFICATION inference.

Inferring: 'HIM' must be a PERSON, because it is the TALKEE
This is a RELATION-CONSTRAINT inference.

Int: (TALKING.2 (I TALKING) (talker = GIRL.2) (talkee = MALE.2))

Relation classification is the simpler of the two classes of nonmarker-passing inferences. The girl is specified as the actor of a talking. FAUSTUS reports that the actor of a talking is called a talker. This is important because there are facts about talkers that are not true about actors in general. However, every actor of a talking is necessarily a talker as a matter of definition. In a similar manner, the patient of the talking is classified as a talkee.

The other class of nonmarker-passing inference is the relation-constraint inference. In the example above, *him* is specified as the patient of a talking event. *Him* maps to the concept *male,* which is defined as an animal whose gender is the male sex, while talking is defined as a kind of communication where the talker and talkee must be people. Therefore, if this male is to be a talkee, he must be a male person, not just a male animal of any kind. FAUSTUS makes the inference that this is the case. Note that these assertions are made during the process of constructing the internal KODIAK representation of the input, which is denoted with the Int: line. After that point markers are passed from each of the concepts in the internal representation, and plausible inferences are suggested.

## 8. STEP 2: PASSING MARKERS

The next step is to pass markers from each of the new concepts in Figure 7 to neighboring concepts in the network. Markers will end up being propagated to a large number of concepts (for the three-line text (6), markers were passed to 279 concepts in a knowledge base of roughly 1,000 concepts). Think of marker passing as spawning new markers which get spread around the network, and not as moving a single marker from node to node.

In most implementations of marker passing, there is a notion of marker *energy* which decays as markers are passed, and prevents markers from spreading indefinitely. Marker energy is typically affected by the type or number of links traversed. Within the paradigm of marker energy, there are various subparadigms. Some researchers have used a system of *energy conservation* where marker energy is divided evenly among the outgoing links (perhaps with some decay or "resistance" to cross the links), and marker

TABLE 2
Path-Half Shapes Defined by Primitive Link Types

| Path Name | Path-Half Shape |
|---|---|
| Elaboration | origin−I−D*−S−C−D*−collision |
| Ref | origin−I−D*−collision |
| Filler | origin−$F^{-1}$−$S^{-1}$−I−D*−collision |
| View | origin−I−D*−V−D*−$C^{-1}$−collision |
| Constraint | origin−I−D*−S−collision |

* Indefinite repetition.
$^{-1}$ Traversal of an inverse link.

passing terminates when the energy reaches zero. Others have used *energy duplication,* where marker energy is propagated across all outgoing links, and is stopped solely by decay, not by the cost of splitting.

In FAUSTUS, the metaphor of marker energy has been abandoned completely, and is replaced by *marker state.* The problem with marker energy approaches is that the criteria for cutting off marker passing is unrelated to the way markers are used. Markers are only useful when they collide, and then only certain collisions are useful; most collisions are spurious. Charniak (personal communication, March, 1985) reports a figure of about 10% useful collisions. Experience with a marker-energy-based marker passer in FAUSTUS corroborates this rough figure, and suggests that the percentage of spurious collisions goes up as the number of concepts goes up. Proponents of marker passing claim that the time necessary to pass markers is irrelevant, because marker passing can be done in parallel on a parallel machine. This is true, but it has been shown here that the evaluation of inferences suggested by marker collisions cannot always be done in parallel. Thus, it *is* cost effective to find ways to cut down on the number of spurious collisions. In attempting to do this, care should be taken to insure that marker passing can still be done in parallel. This means it must be a *localized* operation; it cannot rely on any global information, or there may be memory contention for accessing and/or setting the global state. Before explaining the marker state propagation algorithm, the list of path shapes and inferences recognized by FAUSTUS will be presented.

FAUSTUS has six marker-passing inference classes. Each inference class is characterized in terms of the shapes of the two path-halves which lead to the marker collision. There are give path-half shapes which are defined in terms of primitive link types. These path-half shapes can be described by regular expressions, where a Kleene star (*) marks indefinite repetition, and a($^{-1}$) marks traversal of an inverse link. (See Table 2.) When certain path-halves combine, they form an interesting collision, one that suggests an inference. The six interesting collisions are listed in Table 3; all other collisions are ignored.

TABLE 3
Path-Half Combinations Resulting in Interesting Collisions

| Inference Classes | Path 1 | Path 2 |
|---|---|---|
| Double Elaboration | Elaboration | Elaboration |
| Elaboration | Elaboration | Ref |
| Reference Resolution | Ref | Ref |
| View Application | Constraint | View |
| Concretion | Elaboration | Filler |
| Relation Concretion | Elaboration | Filler |

### 8.1 Is the Marker-Passing Scheme Ad Hoc?

This formulation, like many marker-passing schemes, is open to the criticism of being ad hoc. How are the paths chosen? How are the inference classes chosen? What would happen if one or the other were changed slightly? Such questions must be answered, lest the whole endeavor be considered only a clever programming trick.

The first answer is that the choice of path-halves is solely an implementation decision. Many other choices could be made to achieve the same results (but with varying efficiency). The inference classes, on the other hand, are part of the theory. They could be altered slightly, but the claim here is that any powerful inferencing mechanism must be able to (1) infer relations between objects; (2) establish coreference relations; and (3) establish set membership relations. Here, elaboration and double elaboration for (1), reference for (2), and concretion and relation concretion for (3) are used. There is nothing unusual or ad hoc about these capabilities.

There are minor points which could fruitfully be contested. View application is included as a separate class, while this could perhaps be subsumed under concretion. The two elaboration classes allow for the introduction of a new relation between two existing objects and for the introduction of a new object related (by two new relations) to existing objects, but there is no provision for, say, a new relation between an existing and a new object. So the choice of classes is reflecting a theory of how much evidence is required to make an assumption. This theory can be challenged or modified, but the path-halves are merely a means of implementing the theory, and should not be subject to criticism (unless they contain errors that lead to an incorrect implementation).

### 8.2 The Marker State Propagation Algorithm

In systems like FAUSTUS, which characterize useful collisions in terms of path shape, a *marker state* propagation algorithm is appropriate. The algorithm preprocesses the set of useful path-halves, building a finite state recognizer for the path halves. When a new marker is introduced, it has as its marker state the START node in the finite state recognizer. As markers are

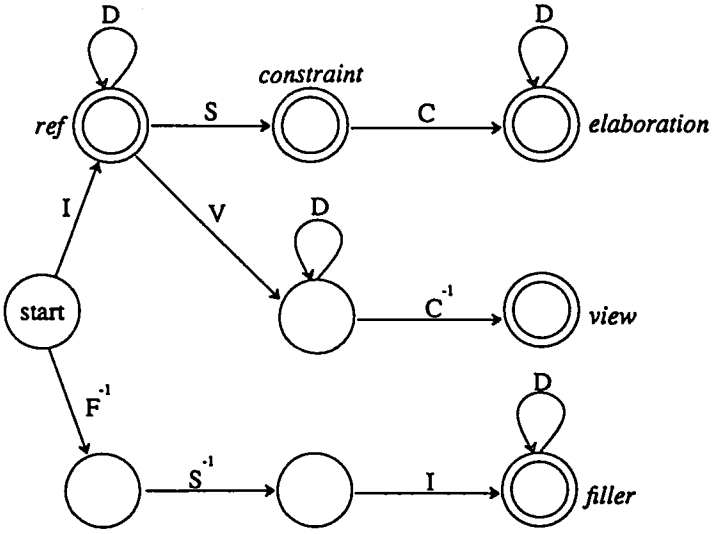


**Figure 8.** Finite state recognizer for FAUSTUS

passed along links from concept to concept they maintain their state. When marker *m* reaches an ACCEPT state *a* at concept *c*, it does two things: (1) adds *m* to *c*'s property list under property *a;* (2) looks under *c*'s property list for other markers that have registered under properties that form a valid path with *a*. These and only these markers indicate useful collisions. Marker passing stops when there are no links to traverse that are in the recognizer. This is where the system derives its advantage: Marker passing can be stopped even when there are outgoing links, if it is known that none of the links are on useful paths.

With a naive algorithm, one would have to inspect $n^2$ collisions for a concept with *n* markers, but using the marker state algorithm, the number can be much smaller. The exact speedup depends on the number of different path-halves and collision types. The finite state recognizer used by FAUSTUS is shown as Figure 8. It has nine states, so the state information could be represented in four bits.

### 8.3 Antipromiscuity Cutoffs

Without some restraints, the marker-passing mechanism as described would still end up checking an enormous number of potential inferences. For example, every situation can have a set of preconditions, which are constrained to be statives, which is dominated by situation. If the marker-passing mechanism is allowed to work unchecked, elaboration collisions would occur for every pair of instances of situation mentioned in the text, each collision suggesting that one is the precondition of the other. The problem is clear:

At a lower level, definitions state what types of stative can be a precondition for what type of action, but these distinctions are lost when going up the hierarchy. They will be discovered again, and most of the potential inferences will be rejected when the matching routines are applied. Unfortunately, that requires a lot of wasted computation. Worse than that is the possibility of introducing improper inferences. This problem is important because one of the design decisions in FAUSTUS was to have a two-step filtering process where not too many unreasonable inferences would be suggested.

This problem is addressed by Charniak (1985), who suggests the *anti-promiscuity* rule: Do not pass markers to concepts that have more than $n$ links attached to them, for some reasonably large value of $n$. This will be referred to here as the *static antipromiscuity* solution. The problem with this approach is that adding new nodes to the middle of the hierarchy can disturb the link counts, and change the computation unpredictably. For example, suppose there is a high-level concept called thing, which dominates 50 different concepts. This is just the type of concept one would like to identify as a promiscuous one. However, now suppose two new concepts are added into the hierarchy just below thing, namely animate-thing and inanimate-thing. Then thing will no longer be promiscuous under Charniak's formulation. It may be that the effect of having thing be promiscuous is achieved if animate-thing and inanimate-thing still have enough links to be promiscuous, and if marker passing only proceeds up the hierarchy. But then the introduction of other intermediate nodes would just push the problem down one level.

Because of this difficulty, the *dynamic antipromiscuity* solution has been adopted here, which works as follows. First, run the algorithm on a representative sample of texts. Then count the markers that accumualate at each concept, and declare the $m$ concepts with the most markers as promiscuous concepts. In this approach, the introduction of new concepts like animate-thing and inanimate-thing will not change the total number of markers that ultimately arrive at thing. Both solutions have an element of arbitrariness; Charniak must choose a value for $n$ and a topology of the network, while here both a value for $m$ and a representative sample of texts must be chosen.

Another change in dynamic antipromiscuity is that passing markers to promiscuous concepts is not stopped, one just stops making certain classes of potential inferences at those concepts. This solution is appealing for several reasons. First, if one is assuming a parallel implementation of marker passing, then there is no cost in continuing to pass markers. Even in a sequential simulation of parallelism, the promiscuous nodes are near the top of the hierarchy, and thus the cost of continuing to spread is not high. The sequential part of the algorithm—sifting through the collisions and evaluating inferences—would be slowed down by a proliferation of markers, so that is where the antipromiscuity rule comes in to play. Spurious elabora-

tion inferences can be ruled out, but other inference classes can still be considered, like referential inferences, which seem to require marker collisions at very high levels in the hierarchy. To see that high-level collisions are sometimes important, consider the first two sentences of text (1), repeated here as (8). The word *it* involves the representation of something like physical-object, or perhaps something even more abstract; *it* can sometimes refer to a situation or an idea. If physical-object were marked as a promiscuous concept, then under Charniak's scheme there would be no way to get the collision that would generate the inference that *it* refers back to the bicycle. In the scheme here, a collision would be detected at a promiscuous concept, but this would be a *referential* collision, a type that allows collisions at promiscuous concepts, if exactly one of the marker origins is explicitly marked as a reference. Reference collisions like the one at person from Bill and John are thrown out because person is a promiscuous concept. Other collision types will be seen that also refuse to suggest an inference if they occur at a promiscuous concept.

(8) Bill had a bicycle. John wanted it.

## 9. STEP 3: SUGGESTING INFERENCES

Associated with each inference class is a suggested inference. The suggestions can add a new concept to the construal of the text, adding a new relation between concepts, construing one object to be coreferential with another, or classifying some concept or relation to be a member of some class. The claim is that these are the only kind of inferences that need be made (for a certain level of understanding of the text), and that the six inference classes are sufficient to generate the appropriate construals.

In story-understanding systems such as Dyer's (1982) BORIS and in most expert system programs, there can be hundreds of inference rules, and adding new knowledge means adding new rules. In FAUSTUS, adding new knowledge is done declaratively, without modifying the basic inference classes. Thus, FAUSTUS' inference classes are very different from the inference rules in traditional expert systems.

Suggestions take the form of entries on an agenda, or queue. Each entry has an inference class and some specific information that depends on the type. For example, the suggestion of finding a referent for a pronoun would have information giving a list of possible referents. When a suggested inference is run, it does one of three things: *succeeds* and builds new representations into the world model, *fails* and removes itself from the agenda without building any representations, or *defers* and puts itself back into the agenda.

The most important part of the algorithm is the set of inference classes, as listed below.

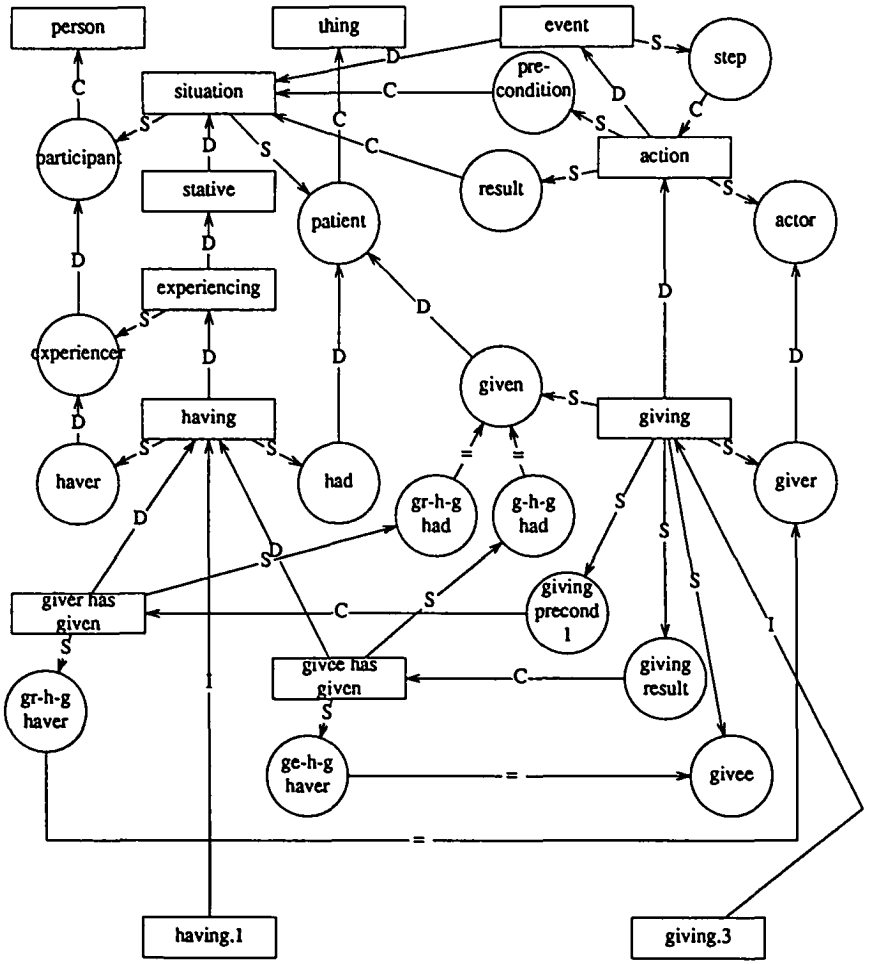


**Figure 9.** Elaboration/referent path collision

## 9.1 Elaboration Inferences

Elaboration inferences are those that build a new relation between two concepts, filling in a slot of some concept with another concept. There are two varieties of elaboration inferences: A collision between two elaboration paths creates a new instance as one of the concepts, while a collision between an elaboration path and a referent path uses existing instances.

When an elaboration path and a referent path collide, the suggested inference is that the concept at the origin of the referent path might be the filler of the slot in the elaboration path. For example, Figure 9 shows a referent path from having.1 colliding at having with an elaboration path

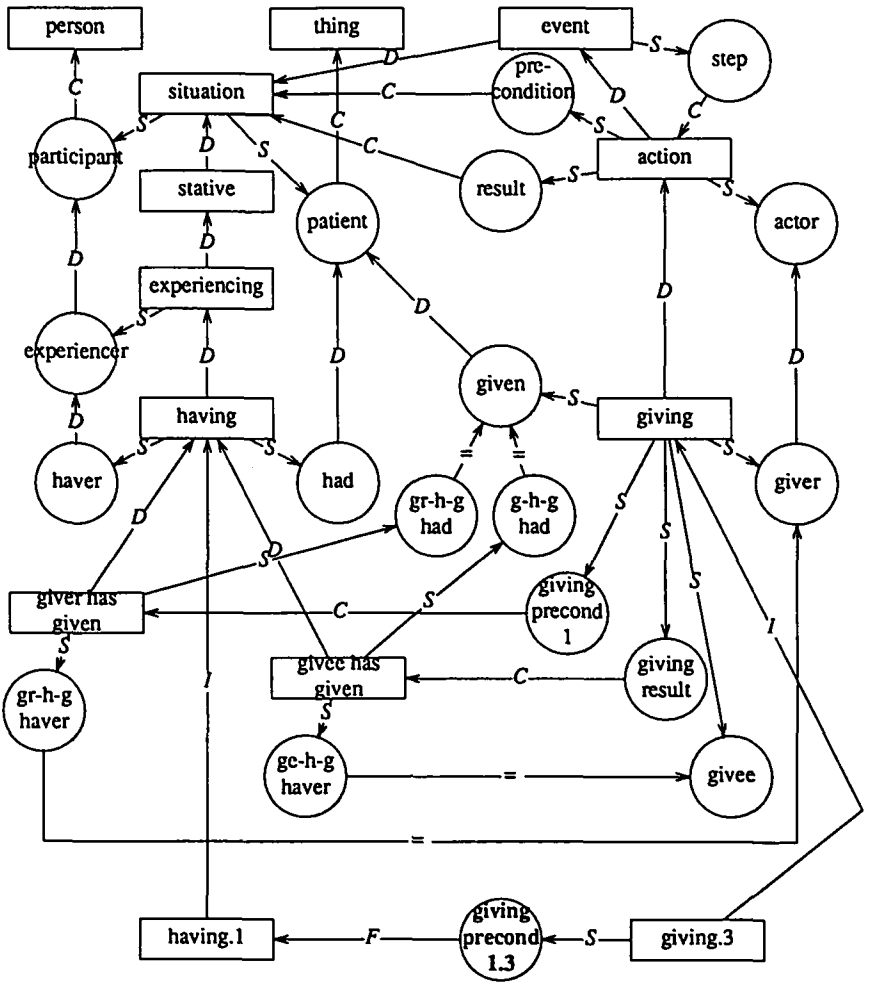


**Figure 10.** Elaboration inference made

from giving.3. The suggested inference is that having.1 might be a precondition of giving.3. In this particular case, the suggestion will be accepted, because there is no evidence to contradict it, and no better alternative filler for that slot. Figure 10 shows the result of an elaboration inference: a new relation, giving-precondition1.3, is built (at the bottom middle of the diagram) indicating that having.1 is in fact a precondition of giving.3.

Not all suggestions are accepted. There is another collision at having suggesting that having.1 is the result of giving.3. This elaboration inference will be rejected when it comes time to evaluate it because the match is not satisfied. There is an after relation that must hold between a situation and its result, but having.1 occurs before, not after giving.3.

The other type of inference involving an elaboration path is the *double-elaboration* inference, when two elaboration paths collide. Although there are no good examples of this in the sample text, it does show up in the fisher-boy story, text (1) above, in the phrase *hoping to catch a few fish from the sea, which they could sell.* The connection between catching the fish and selling the fish is an instance of having the fish. Unlike the previous examples, there is no direct link between the two, and there are no instances of having mentioned in the text. Instead, there is a two-step connection: the having is a result of the catching and a precondition of the selling. One might guess that this difference would require completely different marker paths and inference classes than the case of simple elaborations, but it turns out that the same shape marker paths are involved. Markers pass from catching via result to having-quarry to having, and from selling via precondition to having-merchandise to having. Both of these paths have the shape I→S→C→D, which matches the specification of an elaboration path.

Double elaboration inferences always wait in the suggestion queue before they are evaluated. The reason for this is that the next sentence might provide an explicit filler for the slots, in which case it would be incorrect to make up a new instance for that purpose. The purpose of the agenda is to delay making any decisions until pairs of inferences like these have time to hook up with each other.

Double elaboration inferences solve a problem that Charniak (1972) addressed with a much more complicated approach. Charniak was concerned with texts like the following:

(9) Janet was going to get a present for Jack.
She needed some money.

(10) Janet needed some money.
She was going to get a present for Jack.

In both cases, the reader should infer that Janet needed the money because of getting the present. Charniak's solution was to attach to the concept get-present a procedure, or demon, which would look for an instance of needing-money, and if found, assert that the needing and getting are causally related. As (9) and (10) show, the demon must be able to look both forwards and backwards. The problem with this shows up in cases like (11):

(11) Janet was going to get a present for Jack.
She went to get her piggy bank.

Here, the demon on get-present will not find the need-money it is looking for, and thus can make no inference. But this is not the only demon around. In Charniak's formulation there is also a demon on get-piggy-bank which is also looking for an instance of need-money, and, once found, will assert

that needing money is the reason for getting the piggy bank. To make the right inferences in the case of (11), Charniak must introduce a complicated mechanism called *demon-demon interaction.*

In FAUSTUS, there is no need for demons, and thus there is no need to instruct demons which direction to look, or how they should interact with one another. Instead, one needs only to add to the knowledge base an enables relation between have-money and get-present, and a reason relation between get-piggy-bank and have-money. Once that is done, the double-elaboration inference class will find the right connection in (11), and instantiate the right inferences.

### 9.2 Referential Inferences

Referential inferences allow, among other things, a pronoun or definite noun phrase to refer to a previously represented concept. There are actually three distinct classes of referential paths, depending on the concept at the origin of the path. If the concept is marked indefinite, the path is classified as a *referent* path; if it is marked definite, the path is a *reference* path, and if it is unmarked, it is simply a *ref* path. Noun phrases in English are often marked: "a friend" is indefinite and thus a referent; "the friend" is definite and thus a reference. Pronouns are also considered references. Some noun phrases and most verb phrases are unmarked.

Whenever a reference path collides with a referent path the suggestion is that the reference refers to the referent. (For a ref-ref collision, the suggestion is made both ways.) Suggestions are collected, and stored under each reference. For a given reference, if there is exactly one referent it is accepted, and an equate link is added to show this. If there are several referents, the subset of the referents that match the reference in the highest number of features or relations is found. For example, a boy is defined as a child, a male, and a person, whereas a friend is defined as a person who participates in a friendship relation. Therefore, in "*A boy was talking to a friend. The girl saw him.*" the reference "him" would match the referent "a boy" on two counts, male and person, while it would match "a friend" on only one. Thus, "the boy" would be the chosen referent.

Ties in this counting process are resolved using *recency* and *focus.* Recency is the number of sentences that have passed since a possible referent was last mentioned, and focus is the idea that concepts that have served as slot fillers for salient case slots like actor are more likely to be referents than concepts that were fillers of peripheral case slots, or no slots at all. For example, in "*A boy was talking to his father. The girl saw him.*" the referent for "him" would be the boy, because the boy appears in a focused (actor) slot, while the father does not. Work like that of Grosz (1977) develops the notion of focus much further than this.

There are well-known syntactic contraints on reference. Reflexive pronouns are one source of these constraints, and the notion of C-command (and its variants) is another source. The parser that is currently used does not compute these constraints, but if one were to switch to a parser that did, the algorithm could accommodate this information to rule out certain coreference interpretations. The point is that any reference resolution mechanism should use all the syntactic information available to it, should use semantic and pragmatic constraints, and should fit in with other inferencing mechanisms. The current scheme satisfies those criteria.

How to combine evidence from various sources is an open research question; FAUSTUS uses a fairly primitive counting procedure that does not try to give a good answer to questions like how much focus is necessary to offset one time unit of recency. This mechanism fails to make a reference determination in some cases, and makes intuitively incorrect determinations in other cases. However, it is satisfactory for the vast majority of cases.

One complication is that the surface article in English is not as reliable as one might hope it would be. For example, the article *the* can be used to refer to a specific entity that had not previously been mentioned in the text, as long as it is a well-known entity, such as *the sun* or *the president*. It can also refer to some role in a known script or other type of knowledge structure, as when one refers to *the waiter* in a restaurant. FAUSTUS handles this by finding no referent for *waiter* and then using an elaboration inference to relate the waiter to his role.

Another complication is that some phrases are not marked with any article at all in English. This is particularly true for verb phrases. In passages like (12), the *talked* in the second sentence refers to the same event as the *discussed* in the first sentence, but neither event is explicitly marked as definite or indefinite. FAUSTUS is able to make the inference that the two actions are coreferential, using the same mechanism that works for pronouns. The idea of treating actions under a theory of reference is covered in Lockman and Klappholz (1980). A trace of FAUSTUS processing (12) follows:

(12) The president discussed Nicaragua. He talked for an hour.

```
The President

[1] The president discussed Nicaragua.

Rep: (DISCUSSING (ACTOR = THE PRESIDENT)
                 (CONTENT = NICARAGUA))

[2] He spoke for an hour.

Rep: (TALKING (ACTOR = HE) (DURATION = AN HOUR))

Inferring: 'HE' refers to the PRESIDENT.
   This is a REFERENCE inference.
```

Inferring: the NICARAGUA is a COUNTRY such that
it is the HABITAT OF 'HE' and
it is the COUNTRY of the PRESIDENT.
This is a DOUBLE-ELABORATION inference.

Inferring: the TALKING refers to the DISCUSSING.
This is a REFERENCE inference.

Although this example was meant only to illustrate action/action coreference, there is another inference, where Nicaragua is taken to be both the residency of the president, and the country which he presides over. I did not expect this inference to occur; like most readers, I interpreted the text as referring to the president of the United States. However, this is because I am living in the U.S., and the current U.S. president is a salient figure. FAUSTUS does not have this context available to it. Nowhere was it specified to FAUSTUS that the texts are being read in the U.S., so given that (lack of) context, Inference 3 is quite reasonable.

### 9.3 View-Application Inferences

Sometimes it is necessary to view one concept as another in order to make the right interpretation. An example of this shows up in the sentence *The Red Sox killed the Yankees.* There is a constraint violation in that the Killed-Of-Killing relation should hold between an instance of killing and an animal, but the Yankees are defined as a baseball team, which is an organization, and not an animal. To resolve this constraint violation, we need to interpret either the Yankees as a kind of animal, or the killing as a kind that holds between organizations. Views are applied to make interpretations like this, but they are only applied as needed. Ordinarily, markers are not passed along view links. However, if an input representation contains a constraint violation, then the markers originating at each of the concepts involved in the violation are free to traverse view links.

This view and constraint paths work in tandem much as the referent and reference paths do; there is an inference associated with the intersection of a constraint path and a view path, but neither of them interact with any of the other types of paths. The inference routine associated with their intersection checks to see if viewing the concept at the origin of (17) as an instance of the concept at the collision could rectify the restraint violation that started this passing along view links. If so, the suggested inference is to apply the view. A trace of FAUSTUS processing the kill example follows:

Baseball

[1] The Red Sox killed the Yankees.

Rep: (KILLING (ACTOR = THE RED-SOX) (PATIENT = THE YANKEES))

Inferring: the KILLING is viewed as a DEFEAT-CONVINCINGLY,
where the YANKEES is the DEFEATED of it.
This is a VIEW-APPLICATION inference.

Inferring: the KILLING is viewed as a DEFEAT-CONVINCINGLY,
where the RED-SOX is the DEFEATER of it.
This is a VIEW-APPLICATION inference.

It should be emphasized that the knowledge base already included a mapping between kill and defeat-convincingly; the problem here was to find the proper mapping and apply it to this particular case.

When more than one view is applicable, one has to choose between them. The first step is to see if one of the applicable views is dominated by another. If so, consider only the most specific, and eliminate the more general view from consideration. If this fails, the next step is to defer making a decision for one time unit. On the next time around, it may be that newly inferred links to the concepts involved will enable a choice to be made.

As an example of multiple applicable views, consider the sentence *The chairman moved the meeting up a week.* Moving something up should take as object a direction or distance, not a time duration. To understand this sentence, we need a view that maps time onto a physical direction or distance scale. There are two such views, one that measures distance from the current time into the future, and one that measures time from some landmark in the future back to the current time. Applying the first view would mean interpreting the sentence as meaning the meeting was postponed, while the second view would have the meeting occurring earlier than originally planned. Given both views, FAUSTUS finds both interpretations, but does not choose between them.

There are cases where a view should be applied even when there is no constraint violation. For example, in *Lendl killed Becker at Wimbledon,* there is a valid literal interpretation, but the preferred interpretation still views *killed* as defeat convincingly. Similarly, it is literally true that *no man is an island, entire of itself;* but in processing the Donne quote it would be best to interpret *island* as an isolated-entity, rather than a land-mass. FAUSTUS does not handle such cases.

Jacobs and Rau (1985) used views to handle certain very general relationships in language. The problem he addressed was generating English sentences. Views can be used to handle metonymic relations as well as metaphoric ones, in a manner similar to that described by Hobbs et al. (1988), and adopted by Charniak and Goldman (1988).

Martin (1987, 1988) presents a system called MIDAS (Metaphor Interpretation, Denotation and Acquisition System) which can learn a new metaphorical mapping from input that uses a known, related metaphor. This system can also choose between multiple applicable metaphors, and can find a metaphorical interpretation even when the literal reading is valid.

### 9.4 Concretion Inferences

In the KL-ONE language, much attention is paid to the process of *classification*[1] (see Schmolze & Lipkis, 1983). For example, when given an instance of traveling with an automobile as vehicle, the KL-ONE classifier could conclude the traveling must also be an instance of driving. When given a description of an animal with four legs, a trunk, large ears, tusks, and grey skin, the classification algorithm could conclude that the animal must be a quadruped, but no more. One would like to be able to do more than that, and infer that the animal is probably an elephant. This is a plausible, defeasible inference, not a logical consequence of the taxonomy, and thus is beyond the scope of KL-ONE classification. Such an inference is called a *concretion* inference in FAUSTUS. It refers to the procss of interpreting a concept as something more concrete—less abstract—than is strictly warranted by the representation. Making a concretion inference means adding a dominate or instance link from the concept to a category. Concretion was first discussed in Wilensky (1983) and Norvig (1983b).

To demonstrate how concretion works in FAUSTUS, use the example *John cut the grass.* FAUSTUS infers an instance of lawn-cutting, not just any kind of cutting. This is important, because there are several specific facts associated with lawn-cutting. For example, it is likely that John used a lawnmower as an instrument, and that he cut the blades of grass horizontally, to a roughly uniform height. If we stopped at cutting, and did not make the concretion inference, we could not make the lawnmower interpretation. It would be just as likely that the instrument was a chainsaw, and that he hacked the turf into two large pieces.

Examples of relation concretion are shown in texts (13a–d). The problem is that the preposition *with* is ambiguous. It can mark an accompanier, instrument, manner or just a default "along with/in proximity to" modifier.

(13a) John ate spaghetti with Frank.
(13b) John ate spaghetti with a fork.
(13c) John ate spaghetti with gusto.
(13d) John ate spaghetti with pesto.

For each of these, the parser produces a representation where the relation denoted by "with" is vague. FAUSTUS's job is to concrete the vague relation to one of the known cases. In (13a), the relation-concretion collision at person suggests that the with relation should be concreted to an accompanier. Another collision, at phys-obj, suggests that the with be interpreted as an instrument. In cases where there are multiple possibilities, the evaluation rules favor the more specific interpretation, in this case accompanier. In (13b), only the instrument interpretation is suggested, so it is accepted.

[1] Actually, the term "realizer" was used for this, but I will stick to "classification."

In (13c), neither of these are suggested, since "gusto" is not a physical object. Instead, the manner relation is suggested and accepted.

For (13d) the parser's representation is incorrect; it attaches the prepositional phrase to the verb rather than the direct object. Charniak and Goldman (1988) has shown that marker-passing techniques can be used to decide the proper attachment for prepositional phrases, but FAUSTUS does not address this problem since it is not integrated with the parser, and there is no way to get the parser to produce a representation that is neutral as to attachment. However, when presented with the representation:

```
Rep: (EATING (ACTOR = JOHN) (PATIENT = SPAGHETTI
                (WITH = PESTO)))
```

FAUSTUS is able to make the right inference. The accompanier, instrument, and manner interpretations are not open, since they only hold for actions, and spaghetti is not an action. The only possibility remaining is the default "along with" interpretation. In fact, a more specific version of this relation represents the fact that sauces go with other foods, and given the knowledge that pesto is a sauce, FAUSTUS is able to concrete to the sauce relation.

## 10. Step 4: EVALUATING SUGGESTIONS

As stated earlier, suggestions are placed on a queue called the agenda, and then evaluated after marker passing has been completed for an input. The reason for maintaining an agenda rather than just evaluating each suggestion as soon as it is detected is that some suggestions cannot be evaluated in their own right, but must be compared to other, competing suggestions. For example, in text (1), the bicycle story, there were two competing suggestions: one that 'he' referred to John, and the other that 'he' referred to Bill. Both suggestions are plausible, and either one would be accepted if evaluated separately. Therefore, they must be evaluated together, and the best possibility selected. Evaluation becomes a question of relative plausibility, rather than absolute yes/no acceptance.

All the inference classes offer the possibility of competing inferences. Referents can compete for the same reference, fillers can compete to elaborate the same relation, and multiple possible views or concretions of a concept are possible. The agenda evaluation procedure evaluates competing suggestions together as a group. The procedure also arranges to evaluate all single suggestions before those with competing alternatives. This way all possible information is available when forced to make a choice. An alternative would be to try all possible combinations of suggested inferences, and pick the best resulting state. Unfortunately, the complexity of this approach is exponential.

It is useful to think of the suggestion mechanism as a three-part filter. Out of the infinite number of possible inferences that could be made, completely irrelevant ones are filtered out by only considering inferences due to particular types of marker collisions. Then, out of all the possible inferences due to collisions, impossible ones are filtered out. For example, if 'John,' 'Mary,' and 'he' have been mentioned in the text, then there will be a referential collision with markers originating at 'Mary' and 'he' and colliding at the concept animal. But this will not suggest an inference because 'Mary' and male are in mutually disjoint portions of the hierarchy. Finally, the third filter is a more stringent check of suggestions at evaluation time. A suggestion is made unless there is an explicit contradiction—such as two concepts being in mutually disjoint categories—but the suggestion is only accepted after a full check for contradictions involving all known relations.

## 11. EARLY MARKER PASSING BASED INFERENCES

Now that the FAUSTUS system has been described, it can be compared to previous systems. One of the first inferencing programs was Quillian's (1969) Teachable Language Comprehender, or TLC, which took as input single noun phrases or simple sentences, and related them to what was already stored in semantic memory. For example, given the input "lawyer for the client," the program could output "at this point we are discussing a lawyer who is employed by a client who is represented or advised by this lawyer in a legal matter." The examples given in Quillian (1969) show an ability to find the main relation between two concepts, but do not go beyond that. One problem with TLC was that it ignored the grammatical relations between concepts until the last moment, when it applied "form tests" to rule out certain inferences. For the purposes of generating inferences, TLC treats the input as if it had been just "Lawyer. Client." Quillian suggests this could lead to a potential problem, and presents the following examples:

| | |
|---|---|
| lawyer for the enemy | enemy of the lawyer |
| lawyer for the wife | wife of the lawyer |
| lawyer for the client | client of the lawyer |

In all the examples on the left hand side, the lawyer is employed by someone. However, among the examples on the right hand side, only the last should include the employment relation as part of the interpretation. While a solution in general terms is suggested, Quillian admits that TLC as it stood could not handle these examples.

FAUSTUS has a better way of combining information from syntax and semantics. Both TLC and FAUSTUS suggest inferences by spreading markers from components of the input, and looking for collisions. The difference is that TLC used syntactic relations only as a filter to eliminate certain sugges-

tions, while FAUSTUS incorporates the meaning of these relations into the representation before spreading markers. Even vague relations denoted by *for* and *of* are represented as full-fledged concepts, and are the source of marker-passing. Often a concretion inference will find a more specific interpretation for these vague relations, but if none is found, FAUSTUS will leave the interpretation vague.

Given the input "Lawyer. Client." FAUSTUS can find a connection between lawyer and client without the for relation, just like TLC. Markers originating at the representations of "lawyer" and "client" collide at the concept employing-event. This double elaboration path suggests that the lawyer employs the client.

Given the "lawyer for the enemy" a nonmarker passing inference first classifies the for as an employed-by relation, because a lawyer is defined as a professional-service-provider, which includes an employed-by slot as a specialization of the for slot. This classification means the enemy must be classified as an employer. Once this is done, FAUSTUS can suggest the employing-event that mediates between an employee and an employer, just as it did above. Finally, given "enemy of the lawyer" the of is left with the vague interpretation related-to, so the enemy does not get classified as an employer, and no employment event is suggested.

### 11.1 Script-Based Inferences

The SAM (Script Applier Mechanism) program Cullingford (1978) was built to account for stories that refer to stereotypical situations, such as eating at a restaurant. A new algorithm was needed because conceptual dependency couldn't represent scripts directly. In KODIAK, there are no arbitrary distinctions between "primitive acts" and complex events, so eating-at-a-restaurant is just another event, much like eating or walking, except that it involves multiple agents and multiple substeps, with relations between the steps. Consider the following example:

The Waiter

[1] John was eating at a restaurant with Mary.

Rep: (EATING (ACTOR = JOHN) (SETTING = A RESTAURANT)
(WITH = MARY))

Inferring: a WITH of the EATING is probably
the ACCOMPANIER
because Mary fits it best.
This is a RELATION-CONCRETION inference.

Inferring: the EATING is a EAT-AT-RESTAURANT.
This is a CONCRETION inference.

[2] The waiter spilled soup all over her.

Rep: (SPILLING (ACTOR = THE WAITER) (PATIENT = SOUP)
(RECIPIENT = HER))

Inferring: there is a EAT-AT-RESTAURANT such that
the SOUP is the FOOD-ROLE of it and
the RESTAURANT is the SETTING of it.
This is a DOUBLE-ELABORATION inference.

Inferring: there is a EATING such that
the SOUP is the EATEN of it and
it is the PURPOSE of the RESTAURANT.
This is a DOUBLE-ELABORATION inference.

Inferring: there is a EAT-AT-RESTAURANT such that
the WAITER is the WAITER-ROLE of it and
the SOUP is the FOOD-ROLE of it.
This is a DOUBLE-ELABORATION inference.

The set of inferences seems reasonable, but it is instructive to contrast them with the inferences SAM would have made. SAM would first notice the word *restaurant* and fetch the restaurant script. From there it would match the script against the input, filling in all possible information about restaurants with either an input or a default value, and ignoring input that didn't match the script. FAUSTUS does not mark words like *restaurant* or *waiter* as keywords. Instead it is able to use information associated with these words only when appropriate, to find connections to events in the text. Thus, FAUSTUS can handle the fleeting script problem: Given *John walked past a restaurant,* it would not infer that he ordered, ate, and paid for a meal.

### 11.2 Plan-Based Inferences

In the previous section it was shown that FAUSTUS was able to make what have been called "script-based inferences" without any explicit script-processing control structure. This was enabled partially by adding causal information to the representation of script-like events. The theory of plans and goals as they relate to story understanding, specifically the work of Wilensky (1978), was also an attempt to use causal information to understand stories that could not be comprehended using scripts alone. Consider story (14):

(14a) John was lost.
(14b) He pulled over to a farmer by the side of the road.
(14c) He asked him where he was.

Wilensky's PAM program processed this story as follows: From (14a) it infers that John will have the goal of knowing where he is. From that it infers he is trying to go somewhere, and that going somewhere is often instrumental to

doing something there. From (14b) PAM infers that John wanted to be near the farmer, because he wanted to use the farmer for some purpose. Finally (14c) is processed. It is recognized that asking is a plan for knowing, and since it is known that John has the goal of knowing where he is, there is a match, and (14c) is explained. As a side effect of matching, the three pronouns in (14c) are disambiguated. Besides resolving the pronouns, the two key inferences are that John has the goal of finding out where he is, and that asking the farmer is a plan to achieve that goal.

In FAUSTUS, the same interpretation of the story is arrived at by a very different method; (14a) does not generate any expectations, as it would in PAM, and FAUSTUS cannot find a connection between (14a) and (14b), although it does resolve the pronominal reference, because John is the only possible candidate. Finally, in (14c), FAUSTUS makes the two main inferences. The program recognizes that being near the farmer is related to asking him a question by a precondition relation (and resolves the pronominal references while making this connection). FAUSTUS could find this connection because both the asking and the being-near are explicit inputs. The other connection is a little trickier. The goal of knowing where one is was not an explicit input, but "where he was" is part of (14c), and there is a collision between paths starting from the representation of that phrase and another path starting from the asking that lead to the creation of the plan-for between John's asking where he is and his hypothetical knowing where he is.

The important conclusion, as far as FAUSTUS is concerned, is that both script- and goal-based processing can be reproduced by a system that has no explicit processing mechanism aimed at one type of story or another, but just looks for connections in the input as they relate to what is known in memory. For both scripts and goals, this involves defining situations largely in terms of their causal structure.

### 11.3 Coherence Relation Based Inferences

In this section inferences based on coherence relations are considered, as exemplified by this example proposed by Kay (1981):

(15) A hiker bought a pair of boots from a cobbler.

From the definition of buying one could infer that the hiker now owns the boots that previously belonged to the cobbler and the cobbler now has some money that previously belonged to the hiker. However, a more complete understanding of (15) should include the inference that the transaction probably took place in the cobbler's store, and that the hiker will probably use the boots in his avocation, rather than, say, give them as a gift to his sister. The first of these can be derived from concretion inferences once we have described what goes on at a shoe store. The problem is that we want to describe this in a neutral menner—to describe not "buying at a shoe store"

which would be useless for "selling at a shoe store" or "paying for goods at a shoe store" but rather the general "shoe store transaction." This is done by using the commercial-event concept, which dominates store-transaction on the one hand, and buying, selling and paying on the other. Each of these last three is also dominated by action. Assertions are made to indicate that the buyer of buying is both the actor of the action and the merchant of the commercial-event. The next step is to define shoe-store-transaction as a kind of store-transaction where the merchandise is constrained to be shoes. With that done, the program concludes that a selling involving shoes is a shoe store transaction, that the selling takes place in a shoe store, and the seller is an employee of the store. Another inference is based on a collision at the concept walking. The purpose of boots is walking, and the walking is to be done by the hiker.

## 12. CONCLUSION

The history of text-understanding systems begins with a series of strong-method programs each with algorithms aimed at processing a particular knowledge structure: scripts, plans and goals, thematic abstraction units, and so on. FAUSTUS was built to test two assertions: first, that these knowledge structures are now well enough understood to be put in a common representation formalism, and second, that the resulting knowledge can be processed with a common algorithm that looks for connections between knowledge structures, rather than an algorithm geared towards processing one particular structure.

In a sense, FAUSTUS was an experiment in self-deprivation. The representation language and the basic inference classes were defined early on in the project. Thus, there was never the possibility of throwing in "one more production rule" to add a special case to account for some annoying bug. Instead, the bulk of the work was in designing good representations of the domain concepts. Because most concepts are used in several texts, and the same knowledge base had to work for all texts, the author was forced to confront representational problems rather than add special cases. The result was a better understanding of the knowledge structures, and a proof-by-example of the two assertions. Thus, FAUSTUS can be seen as a synthesis of much work in the 70s and early 80s. It is unified in that it handles many knowledge structures with one algorithm, and it is extensible in that new types of structures can be added.

Marker passing proved to be a valid implementation technique, and fit well with the metaphor of finding connections between pieces of knowledge. The marker state propagation algorithm is an important addition to the set of marker-passing techniques. However, marker passing was not crucial to the project. Alterman's (1985) NEXUS, for example, used breadth-first

search to find similar connections. Any search method could have been substituted for marker passing as long as it was capable of finding elaboration, concretion, reference, and view inferences (or their equivalent in some other formalism).

FAUSTUS is quite strong at suggesting plausible, relevant, and easy inferences, and at deciding among competing suggestions. It also begins to address a much harder problem: combining evidence from several suggestions, and choosing the best subset of suggestions. This is the primary problem addressed by modern weak-method approaches like those of Charniak and Goldman (1988), Hobbs et al. (1988), Pollack and Pereira (1988) and Stallard (1987). In each of these systems, the idea is to use abduction to pick the most likely or most coherent interpretation. Each system uses some kind of global metric that allows different sources of evidence to be combined in comparing one interpretation with another. The difficulty is twofold: defining the necessary knowledge, with the right metrics, and limiting the search in some way, since an unconstrained search would be exponential.

FAUSTUS constrains search in a very severe fashion: It is deterministic. Like Marcus' (1980) parser, it never backs up once it has accepted an inference. It is able to do this because it keeps a list of possible inferences (the agenda), and carefully selects inferences without competition first. The order of evaluation attempts to account for useful interactions of constraints, without having to consider all possible combinations of inferences exhaustively. Of course, there may be situations where the possible inferences are so deeply intertwined that it is impossible to pick one without considering all others. Also, many texts are interesting precisely because they induce the reader to infer one thing, and then reveal that in fact the opposite is true. FAUSTUS is unable to handle these texts. Thus, it is a tribute to the coherence of texts, and not to FAUSTUS' combinatorial powers, that FAUSTUS is able to handle as many texts as it does.

FAUSTUS also suffers from being an isolated module. It is not integrated with the parser; problems stemming from this are well known. Another problem is that the suggestion-finding mechanism is not really integrated with the suggestion-evaluation mechanism. FAUSTUS has no notion of a coherent or noncoherent text: It gathers suggestions and accepts as many of them as possible. It does not notice if there are "too few" inferences to make the text coherent.

The future of text-processing systems seems to lie with the abduction paradigm. More powerful systems will need better and more complete representations of knowledge, better integration with neighboring modules, and a way of constraining search so that probable interpretations can be abduced in tractable time.

■ Original Submission Date: August 9, 1988

## REFERENCES


Alterman, R. (1985). A dictionary based on concept coherence. *Artificial Intelligence, 25,* 153–186.

Anonymous (1972). *My big book of fairy stories.* Manchester, England: G.G.S. Ltd.

Brachman, R.J., & Schmolze, J.G. (1985). An overview of the KL-ONE knowledge representation system. *Cognitive Science, 9,* 171–216.

Charniak, E. (1972). Toward a model of children's story comprehension. (AI Tech Rep. No. 266) MIT AI Labs, Cambridge, MA.

Charniak, E. (1985). A single-semantic-process theory of parsing. Unpublished manuscript. Dept. of Computer Science, Brown University, Providence, RI.

Charniak, E., & Goldman, R. (1988). A logic for semantic interpretation. *Proceedings of the 26th Meeting of the ACL* (pp. 87–94). State University of New York, Buffalo.

Clark, H.H. (1975). Bridging. In R.C. Schank & B. Nash-Webber (Eds.), *Proceedings of TINLAP.* Cambridge, MA.

Cullingford, R.E. (1978). Script application: Computer understanding of newspaper stories. (Research Report. No. 116) Computer Science Dept., Yale University, Boston.

Dyer, M.G. (1982). In-depth understanding. (Research Report No. 219). Computer Science Dept., Yale University, New Haven, CT.

Fahlman, S.E. (1979). NETL: A system for representing and using real-world knowledge. Cambridge, MA: MIT Press.

Grosz, B. (1977). The representation and use of focus in dialogue understanding. Doctoral dissertation. University of California at Berkeley.

Hobbs, J.R., Strickel, M., Martin, P., & Edwards, D. (1988). Interpretation as abduction. *Proceedings of the 26th Annual Meeting of the ACL.* State University of New York, Buffalo.

Jacobs, P.S., & Rau, L.F. (1985). Ace: Associating language with meaning. In *Advances in Artificial Intelligence* (pp. 295–304). Amsterdam: Elsevier North Holland.

Kay, P. (1981). *Three properties of the ideal reader.* Berkeley Cognitive Science Program, Berkeley.

Lockman, A., & Klappholz, A.D. (1980). Toward a procedural model of contextual reference resolution. *Discourse Processes 3,* 25–71.

Marcus, M.P. (1980). *A theory of syntactic recognition for natural language.* Cambridge, MA: MIT Press.

Martin, J.H. (1987). Understanding new metaphors. *Proceesings of the 10th International Joint Conference on Artificial Intelligence* (pp. 137–139). Milan.

Martin, J.H. (1988). *A computational theory of metaphor.* Doctoral dissertation. University of California at Berkeley.

Norvig, P. (1983a). Frame activated inferences in a story understanding program. *Proceedings of the International Joint Conference on Artificial Intelligence–83* (pp. 624–626). Karlsruhe, West Germany.

Norvig, P. (1983b). Six problems for story understanders (pp. 284–287). *Proceedings of the International Joint Conference on Artificial Intelligence–83,* Washington, DC.

Norvig, P. (1986). A unified theory of inference for text understanding. Doctoral dissertation. University of California at Berkeley.

Pollack, M.E., & Pereira, F.C.N. (1988). An integrated framework for semantic and pragmatic interpretation (pp. 75–86), *Proceedings of the 26th meeting of the ACL.* State University of New York, Buffalo.

Quillian, M. (1969). Ross, The teachable language comprehender: A simulation program and theory of language. Communications of the ACM, *12,* 459–476.

Schank, R.C., Goldman, N., Rieger, C., & Riesbeck, C.K. (1973). Margie: Memory, analysis, response generation and inference in English. *Proceedings of the Third International Joint Conference on Artificial Intelligence* (pp. 255–261). Stanford, CA.

Schmolze, J.G., & Lipkis, T.A. (1983). Classification in the KL-ONE knowledge representation system, (pp. 330–332). *Proceedings of the 8th International Joint Conference on Artificial Intelligence.* Karlsruhe, West Germany.

Stallard, D. (1987). The logical analysis of lexical ambiguity. *Proceedings of the 25th Annual Meeting of the ACL* (pp. 179–185). Stanford, CA.

Wiensky, R.W. (1978). *Understanding goal-based stories.* Yale University Computer Science Research Report, New Haven, CT.

Wilensky, R. (1983). *Planning and understanding.* Reading, MA: Addison-Wesley.

Wilensky, R. (1986). Some problems and proposals for knowledge representation. (Report No. UCB/Computer Science Dept. 86/294). Computer Science Division, University of California at Berkeley.