Knowledge-Based Natural Language Understanding: A AAAI-87 Survey Talk

Wendy G. Lehnert

COINS Technical Report 88-02

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

KNOWLEDGE-BASED NATURAL LANGUAGE UNDERSTANDING

Dr. Wendy G. Lehnert

Department of Computer and Information Science

University of Massachusetts

Amherst, MA 01003

1 Introduction

This overview is organized within a historical framework, although time limitations have forced me to invent a version of history that is necessarily incomplete. The title of the talk was given to me by the AAAI Program Committee, who wisely restricted the scope of my task by including the descriptor "knowledge-based." This mercifully allowed me to ignore a large body of work that focuses exclusively on the syntactic structures of natural language. Even so, the body of work that can accurately be described as "knowledge-based natural language understanding" is large, and difficult to cover in the space of one hour. To maintain continuity, I utilized the recurring theme of weak methods vs. strong methods. This foundational theme helped me pare down my view of history and serves as my only defense against otherwise unforgivable omissions in the overview. Even so, it was difficult to pick and choose from the corpus of potentially relevant research, and the usual disclaimers about intelligible brevity at the cost of comprehensive coverage must be piously invoked to ward off inevitable accusations of ignorance, prejudice, and other sins associated with warped thinking.

I'm going to use a lot of examples to illustrate key concepts, interleaving the examples with a chronological survey of the literature. We'll periodically try to rise above the trees to see the forest, and search for threads of strong methods and weak methods throughout. We'll see how strong methods came to dominate the field for a period of time, only to be followed by the pendulum's swing toward weak methods, where we seem to be today.

If we go back to the beginning of time, we go back about 15 years. I would date 1972 as a convenient starting point for knowledge-based natural language processing. There were two very important pieces of work that surfaced around 1972. First, Terry Winograd published his Ph.D. dissertation under the title *Understanding Natural Language*. [Winograd 1972]. At the same time, Eugene Charniak completed his Ph.D. dissertation on a model of children's story comprehension. [Charniak 1972] Both of these theses came out of MIT - in fact, Charniak and Winograd were office-mates at MIT.

Despite the physical proximity of the authors at the time, these two views of natural language processing couldn't be more different. Let me read you an excerpt from a recently published retrospective by Terry Winograd. In his own words, he sums it up as follows:

> "Fifteen years ago, a program named SHRDLU demonstrated that a computer could carry on a simple conversation about a blocks world in written English. Its success led to claims that the natural language problem had been solved and predictions that within a short time conversations with computers would be just like those with people.
>
> ... With years of hindsight and experience, we now understand better why the early optimism was unrealistic. Language, like many human capabilities, is far more intricate and subtle than it appears on first inspection." [Winograd 1987]

That's Terry Winograd speaking in 1987. To understand the significance of his cautionary hindsight, we must first understand that there was tremendous excitement over SHRDLU when it was initially publicized in the early 70s. There was much less excitement over Charniak's relatively unknown thesis, although we do find people referencing it even now. Hubert Dreyfus, a well-known professional critic of AI, says the following about Charniak:

> "... by 1970, AI had turned into a flourishing research program, thanks to a series of microworld successes, such as Winograd's SHRDLU, Evan's Analogy Problem Program and Winston's program which learned concepts from examples.

... Then rather suddenly, the field ran into unexpected trouble. It started, as far as I can tell, with the failure of Charniak's attempts to program children's story understanding. It turned out to be a much harder problem than one expected to formulate a theory of common sense. It was not, as Minksy had hoped, just a question of cataloging a few hundred thousand facts." [Dreyfus 1987]

To sum up, Winograd was dealing with a view of language which was very optimistic and designed to convince the world that natural language processing was a viable research problem. Charniak was taking a somewhat more unpopular but realistic stand in looking at the really hard problems we would eventually have to tackle if we were to deal with language in any truly general sense. To digress for a moment, I would like to mention something ironic about Winograd and Charniak. While Charniak was clearly the pessimistic foil to Winograd's optimist, it is amusing to note that Charniak remains extremely active and productive in the field of natural language processing, whereas Winograd has ceased to make contributions to AI, opting instead to investigate the philosophical implications of hermeneutics [Winograd and Flores 1986].

We will look at Charniak's thesis just long enough to note the general emphasis in that research. Here's a quote from the dissertation abstract:

"An earlier version of the model described in this thesis was computer implemented and handled two story fragments, about a hundred sentences. The problems involved in going from natural language to internal representation were not considered, so the program does not accept English, but an input language similar to the internal representation is used." [Charniak 1972]

To be blunt, Charniak's program never analyzed sentences. In some sense, Charniak's thesis was not a thesis about language analysis at all, although I view it as a milestone thesis for knowledge-based language understanding. Charniak was looking at a set of problems which are not specific to sentence analysis *per se*, but nevertheless key to understanding natural language. Charniak was concerned with the problem of inference. That concern evolved into a driving motivation for much of the research on knowledge-based natural language processing we've seen over the last 15 years.

It is useful to contrast the two veins of research that were more or less initiated by Charniak and Winograd. There is ***problem-driven*** research and there is ***technology-driven*** research. I'll characterize problem-driven research as basic research designed for the long haul: given the difficulties inherent in understanding language, what techniques might be of use to us in surmounting these difficulties? Technology-driven research is the research of near-term applications: given the current state-of-the-art, what applications are appropriate for the existing technologies?

SHRDLU was a wonderful example of technology-driven research. The blocks world lent itself to techniques that were available at the time. But SHRDLU was just a prototype designed to inspire further work. The contemporary offspring of that inspiration are found today in database query interfaces. We have a technology-driven research program on natural language interfaces which works (more or less), but is successful primarily because it does not need to deal with natural language in its full generality.

To appreciate the problems of natural language in general, we have to understand what is meant by the inference problem in natural language - the problem that made Charniak such a pessimist about life outside the blocks world. Let's take an example of a short narrative to illustrate the problem:

"When the balloon touched the light bulb, it broke. This caused the baby to cry. Mary gave John a dirty look and picked up the baby. John shrugged and picked up the balloon."

This is a typical example of narrative text. We can analyze it in terms of its information content by distinguishing explicit information from implicit information. We are explicitly told about seven events in this story and one explicit causal relationship signaled by the verb "caused." But implicitly, there's more information. There are at least six implicit events and states that are present in the paragraph, eight implicit causal relationships, and six implicit goal states or emotional states. (See figure 1).

[insert figure 1 about here]

For example, probably the balloon was inflated. Probably the balloon exploded when it broke. There is an ambiguity associated with the pronoun when we are told "it broke." Was it the balloon that broke or the light bulb that broke? Most readers have no trouble understanding that the balloon broke. Furthermore, we might conjecture that the light bulb was on and it was the heat from the light bulb that broke the balloon. These are all plausible common-sense inferences people are able to make. But these are only assumptions and they are assumptions that could be wrong. We will define an inference to be an assumption that could be wrong. Technically speaking, this type of inference is known as defeasible inference, but for the remainder of this talk we'll just call them inferences.

Charniak's interest in children's stories was centered on the problem of inference generation. Children are capable of highly sophisticated inferences which make children's stories extremely complicated for computers. Although the language in children's stories may be relatively simple in terms of syntax and vocabulary, the underlying processes of inference required to understand a typical children's story are not so easy to characterize. The basic problem has to do with knowledge about the world. Children have a great deal of knowledge, although the magnitude of this underlying knowledge base is largely unappreciated by people who have never tried to get a computer to operate with comparable facility.

The general problem of inference generation inspired a lot of work in the mid-to-late 70s devoted to identifying knowledge structures that could spawn inferences. During this period, we saw progress that I would characterize as work in strong methods for natural language processing. By this I mean to say that there was a strong preoccupation with specific knowledge structures and knowledge-specific mechanisms of inference generation. We will briefly outline the major contributions of that period since the work was highly influential, not only within the AI community, but within cognitive psychology as well. (Eventually, we will get around to looking at problems of sentence analysis *per se*.)

2 Knowledge Structures

The first knowledge structure that was proposed as a powerful device for inference generation was the script [Schank and Abelson 1977]. Scripts have trickled down into the introductory textbooks on AI, but if you're not familiar with the concept, I'll run through it very briefly.

Scripts are designed to encode stereotypic event sequences. This is mundane knowledge about some standard scenario for which a common linguistic community shares knowledge. So, for example, we all have knowledge about going to the movies. And if I say to you, "I went to a movie last night," you are capable of generating a lot of inferences about what I did last night which go far beyond the explicit information content of that sentence. You understand that I must have had money to buy a ticket and the ticket was purchased at the theatre. I may have had to wait in line for a bit before I could go into the theatre, but once inside I could have bought popcorn, candy, or ice cream. I exchanged the ticket with an usher who gave me a stub back ...

You have all these little facts about going to the movies. These are all assumptions that could be wrong. But for the most part, these are the assumptions you have to make. And if we want to create computers that can understand language, we have to worry about creating systems that generate these inferences as well. This is the implicit information content underlying language.

A system called SAM was first implemented in 1975, which was given simple narratives and then tried to generate inferences appropriate for those stories on the basis of scripts [Cullingford 1978]. SAM stood for "Script Applier Mechanism." The architecture of SAM was fairly simple. There was a parser that mapped sentences into an internal memory representation, in this case, Conceptual Dependency [Schank 1975]. Then the actual script applier mechanism accessed the appropriate scriptal knowledge structure and tried to fill in any missing implicit events in a causal chain representation. "I went to a movie last night," would be expanded into a very long causal chain representation containing all the implicit events associated with knowledge about movies.

SAM was a prototype program designed to demonstrate the utility of one particular knowledge structure. That knowledge structure became somewhat controversial in terms of its generality. Where do scripts work? Where don't they work? Are they appropriate for generating all the inferences we need?

If we go back to our balloon story, we could, for example, hypothesize the existence of a balloon script. Here is our stereotypic event knowledge about balloons: They start out in an uninflated state. They get inflated in one of two stereotypic manners, they get tied, and then they die a natural death in one of three ways (see figure 2).

[insert figure 2 about here]

This is event-oriented knowledge about balloons. If we wanted to understand our little story about the light bulb and the balloon using 1975 technology, we would simply match the explicit input against the events described in the balloon script, and infer that the balloon was inflated and tied before it broke. While these are undeniably nice inferences to have, we wouldn't know anything about why the balloon broke or why it was reasonable for it to break. Indeed, if our "light bulb script" included breakage as one of the stereotypic ways that light bulbs come to an end, there would be no way of knowing which referent (for "it") was broken on the basis of these scripts alone.

At the same time that scripts were being proposed by Roger Schank at Yale, Schank also understood that scripts were not the solution to all of the problems of knowledge based inference generation. He proposed other knowledge structures as well. For example, there was knowledge about plans and goals.

If I told you I hired someone to clean my house, you could make a number of inferences about exactly what that entailed. I had to find someone who would be willing to clean the house, I had to approach this person, ask them to clean my house, there was probably some negotiation over payment, and so on and so forth. All of these inferences are very general in the sense that they would apply to anyone I might hire to do a periodic task for me, such as mow my grass or do my shopping for me. Any number of tasks that keep popping up over and over again could be handled in the same manner. So these inferences appear to originate from a more general understanding of plans and goals. In this case, we have a problem of goal subsumption (finding a solution to a recurring goal), and a solution in terms of agency (locating an agent who will do the work for me). So plans and goals involve a level of abstraction that goes beyond scripts, but which still allows us to characterize stereotypic situations [Wilensky

1978].

A well-known book came out in 1977 which put down in writing all of the ideas that were floating around Yale at that time [Schank and Abelson 1977]. This was a book about knowledge structures, more specifically, scripts, plans and goals, among other things. It was a seminal piece of work insofar as it generated, by my count, ten Ph.D. theses in AI (there were probably a comparable number of Ph.D.s in psychology as well). So there was a tremendous amount of work along these lines in the mid and late 70s, and that work created a foundation for the more recent research to which we now turn.

First, we'll look at two different directions that took off after that initial foundation in knowledge structuring was first laid. In so doing, we'll see different knowledge structures: 1) plot units [Lehnert 1981], and 2) thematic affect units [Dyer 1983b], both of which were designed to produce summaries for narratives.

In both systems, we assume that multiple levels of memory representation are being generated in response to the input text. Sentences are translated into Conceptual Dependency, and inferences are generated via script application and the analysis of plans and goals. In the case of plot units, additional levels of abstraction are required to produce an affect state map, and finally a plot unit graph. The plot unit graph rests on top of all these "lower" levels of memory representation which act, in turn, as conceptual scaffolding for the narrative summarization task.

In the tradition initiated by Charniak's thesis, most experiments run on plot units require hand-coded memory representations at the lower levels in order to see anything of interest at the level of a plot unit graph. Granting that, there is a program called PUGG (the Plot Unit Graph Generator) which generates memory representations of the sort found in figure 3.

[insert figure 3 about here]

This is a plot unit graph generated in response to Arnold Toynbee's synopsis of the New Testament [Alker, et a. 1975]. Note that this graph could never be generated automatically from the source text of the New Testament, given the current state of the art. Just the hand coding of the knowledge structures would necessitate sacrificing an entire generation of graduate students in an orgy of exploitation normally unheard of outside the biological sciences.

Each node in this graph represents an instantiated plot unit where plot units describe things like competition between two characters, or one character's successful resolution of a problem situation. Arcs are created between nodes when two plot units depend on a shared component from the affect state map. In this way, the plot unit graph provides a picture of the conceptual connectivity across the narrative. Ideally, this graph will allow us to identify the salient and most central concepts by looking at the topological features of the graph. For example, the cut points in this graph are very important plot units for our story. The three major cut points for the main body of this plot unit graph point to the following events from the New Testament:

(7) Jesus called on the people to support him.

(47) The authorities arrested Jesus.

(89) The authorities crucified Jesus.

If we wanted to produce a truly minimalist synopsis of the New Testament, we are perhaps on the right track here, although we do not have the explanatory power to tie these three events together into a truly self-contained blurb about Jesus.

We could elaborate on this skeleton a bit by invoking a minimal path algorithm to connect our three cut points. These produce the following event-summary:

(7) Jesus makes an appeal to the masses for support.

(9) The government wants to maintain authority over the masses.

(10) Jesus causes a scandal.

(18) Jesus takes the law into his own hands to avenge God.

(47) The authorities arrest Jesus.

(89) Jesus is crucified.

(92) Jesus' death is a triumph.

(93) Jesus is worshipped.

I am told that this is, in fact, a Marxist interpretation of the New Testament.

Let us now return to the other line of work on narrative summarization that relied on scripts, plans and goals. As we saw with plot units, it is possible to produce narrative summaries based on event descriptions alone, as long as you can identify the central events of the story. But there are other kinds of summaries that operate on a more abstract level of understanding. Fables are famous for the adages associated with them, and the ability to associate an appropriate adage with a novel narrative is considered a hallmark of mature intelligence (understanding the meaning of proverbs is a task used by the Stanford Binet IQ test as a standard for measuring adult intelligence).

Research on thematic affect units addressed this aspect of narrative summarization [Dyer 1983a]. Dyer claimed that adages are properly associated with abstractions at the level of plans and goals. Each thematic affect unit describes a pattern of plan-oriented behavior, and if all the required components of the pattern are met, the specific adage associated with that thematic affect unit will apply.

So for example, a close call, which would perhaps be described by the adage, "a miss by an inch is as good as a mile," could be recognized via the following thematic affect unit:

(1) X experiences a major preservation goal, G.

(2) G was created in response to an event not intended by X.

(3) G is a fleeting goal so no recovery plan is required.

Note that a close call can be easily transformed into a regrettable mistake (don't cry over spilt milk) if G is not characterized as a fleeting goal and a recovery plan therefore becomes appropriate.

It is interesting to note that a plot unit analysis can be performed without the benefit of thematic affect units, and thematic affect units can be recognized without any of the effort associated with affect state maps and plot unit graphs. These two approaches to narrative summarization are fully independent of one another and simply reflect different types of summarization tasks. As far as the computational models are concerned, skills with one task do not predict seemingly associated skills in the other.

Plot units and thematic affect units both emerged from a large research effort centered around a system named BORIS [Lehnert, et al. 1983]. BORIS attempted to integrate a large number of knowledge structures in a single system, addressing the architectural problems posed by multiple knowledge structures. The BORIS system, completed in 1982, marks the end of the knowledge structuring era. For the most part, people stopped proposing new knowledge structures at about that time, and interests shifted into other areas.

To understand why, we need only look at the diagram in figure 4 (taken from [Dyer 1983a]).

[insert figure 4 about here]

BORIS attempted to integrate no less than 22 different knowledge structures, each responsible for generating its own class of inferences encoded with structurally-specific knowledge representations, and using its own structure-specific inference mechanism. Figure 4 tells us what lines of communication were open between the various knowledge structures. Each node of the graph represents a generic knowledge structure, and each arc tells us when one knowledge structure was allowed to talk to another one. Rather than having all possible pairwise channels of communication open, we limit communication between knowledge structures and impose some order on the potential chaos that would otherwise break loose.

Unfortunately, the rich diversity of the knowledge structures requires unique forms of communication between sanctioned pairs of knowledge structures. No two arcs in this diagram are quite the same in terms of the type of information being requested or the methods of computation required to produce a response. Not only are there inference processes specific to each knowledge structure, but the communications between pairs of knowledge structures are pairwise specific.

However impressive BORIS may have been as a tour de force in knowledge-based natural language understanding, the word "elegant" has never graced any noun phrase describing the flow of control in BORIS. "Ad hoc" was rather closer to the truth, and the difficulties of continuing on in this vein were apparent to all. Suffice to say, no one ever attempted to re-implement the BORIS system after Dyer completed his noteworthy thesis based on the system, and no one associated with the original BORIS system went on to produce a son of BORIS. The complexity of the architecture, the fragile scaffolding needed to make it all hang together, and the methodologically difficult business of engineering mundane knowledge for natural language were all overwhelming. Although Dyer has never been accused of being a pessimist, his thesis, published 10 years after Charniak's, was another milestone destined to send the faint-hearted elsewhere in search of smoother sailing.

I think a lot of people realized the implications of BORIS in 1982. Although there was no way to walk away from the need for knowledge, the growing commitment to knowledge-based natural language processing gradually shifted into a wistful longing for processes operating over uniform knowledge representations, inference mechanisms that transcend individual knowledge structures, and elegant control mechanisms that can be explained within the confines of a single page. Of course, there were always people in the field who felt compelled by these aesthetic criteria: Winograd was involved in the development of KRL [Bobrow and Winograd 1977], and even Charniak once described himself as a methodological "scruffy" with a "neat" struggling to get out.[1]

3 Marker Passing

The excitement associated with PROLOG in the early 1980's, and the more recent fever surrounding connectionism, have both exerted a predictable pull over researchers in knowledge-based natural language processing who felt a need to swing the pendulum back a bit from the strong methods associated with wildly propagating knowledge structures. At this time we seem to be swinging back in the direction of weak methods, with a clear question to be answered: does the commitment to knowledge-based techniques necessarily force us into a technology dominated by strong methods? Ten years ago the answer was maybe. Today we seem to be

[1]see (Abelson 1981) for the official explanation of "scruffy" and "neat" as technical terms referring to methodological styles.

saying maybe not.

In keeping with this general trend, we are seeing new work on homogeneous inference generation. The roots for this do go back, so we should take a little time to give credit where credit is due. Probably the earliest reference is Quillian, who first promoted the idea of intersection search in a computational framework. This was followed up by Rieger's thesis work, for which Rieger was honored by being asked to give Computers and Thought Lecture at the 1975 IJCAI. Let me talk a little bit about all of that so we can appreciate the significance of more contemporary contributions to homogeneous inference.

The idea of an intersection search is fairly simple. Quillian is generally credited with the earliest description of an intersection search algorithm [Quillian 1968], but we'll introduce the idea in the context of Rieger's thesis because Rieger's work is more on-target with respect to inference generation [Rieger 1974].

Suppose we have a meaning representation for sentence S1, and a meaning representation for a second sentence, S2. These two representations serve as input to Rieger's program, MEMORY. Each meaning representation then generates a first generation of immediate inferences, which will each recursively spawn a second generation of inferences, then a third generation, and so forth and upward and onward (gee whizz!) [Geisel 1950]. In theory, we can produce inferences arbitrarily far away from the original input sentences.

In an intersection search, this recursive generation of inferences halts when we find a path of inferences connecting the two input generators. If MEMORY can find a path of inferences which starts at S1 and concludes at S2, then we have a good candidate for a causal chain between the two sentences. That is, we have a string of causally connected events and states that take us from one sentence to the next. So we might understand, for example, if the balloon touches the lightbulb (S1) and the balloon subsequently breaks (S2), then there is a causal chain going from (S1) the balloon coming into contact with the lightbulb, to (S1′) the balloon coming into contact with a light bulb that is turned on, to (S1″) the balloon coming into contact with a light bulb that is turned on and hot, to (S2‴) the balloon coming into contact with a hot object, to (S2″) the balloon being in contact with a hot object, to (S2′) the balloon exploding as a result of contact with a hot object, to (S2) the balloon breaking. If an intersection can be established between S1″ and S2‴, we will have a causal chain analysis of the two sentences.[2]

When Rieger employed intersection search for inference generation back in the early 70s, he was not working in a knowledge-based framework. Consequently, there was no knowledge in MEMORY - certainly nothing we would recognize today as a declarative knowledge structure. Rather, Rieger had 16 inference "molecules" that were responsible for the propagation of inferences underlying the intersection search. If there was any knowledge in MEMORY at all, it had to be buried inside the lisp code that realized these 16 inference classes. But in fact, most of the inferences that MEMORY generated were based on simple manipulations of Conceptual Dependency event and state descriptions, and none of those manipulations were dependent on structures outside of the search space being generated during the intersection search. Despite its name, MEMORY had no long-term memory, and the expanding circles of inference it generated were basically pulled out of thin air (or at least 16 thin inference molecules).

If Rieger's thesis looks weak from the perspective of knowledge-based systems, we must remember that he intended to make a contribution regarding search. Indeed, he had an elegant idea concerning the relationship between inference generation and causal chain construction: the

[2] In fact, Rieger's meaning representation language (Conceptual Dependency) was not well suited for this particular example, and MEMORY probably couldn't have found this causal chain, but we're just trying to illustrate the general idea.

construction of a causal chain was a search problem and the undirected generation of inferences created the search space in which to operate. Both components were nicely addressed within the simple framework of an intersection search. This emphasis on the algorithm for search created a model about control, and the beauty of MEMORY's control was its simplicity and homogeneous generality.

Rieger's work is important for us because it illustrates a weak method for inference generation based on a simple mechanism of great generality. We should also note that Roger Schank was Rieger's thesis advisor, and Schank has said that his work on scripts was strongly motivated by what he perceived to be the fatal flaw in Rieger's MEMORY: a lack of knowledge. In Schank's view, the real problems were inside those inference molecules (or whatever mechanisms were needed to generate inferences). The key problem must be to understand the organization of knowledge needed to create inferences. MEMORY was appealing, but sadly predicated on the wrong framework for the problem of inference generation. If inference generation is essentially a problem of search, then MEMORY should give us some answers worth pondering. But if inference generation is better characterized as a problem of knowledge application, then MEMORY must fall very short of the mark. If Rieger made a mistake, it was in asking the wrong question more than in finding the wrong answer.

Now we can move the clock up to 1987 and look at a program called FAUSTUS which identifies 7 classes of inference and activates selected concepts throughout a potentially large search space in an effort to identify useful inferences [Norvig 1987]. At first glance, this may look like a reincarnation of Rieger, but we need to look a little closer. First we note that the simple intersection search has been replaced by a more sophisticated marker passing algorithm. The new algorithm looks like a step in the right direction (it narrows the potential search space), yet we still have homogeneous control for inference generation. How is this possible?

It seems that FAUSTUS benefited from all the work that followed and superceded Rieger without sacrificing the weak method of homogeneous control. FAUSTUS utilizes extensive amounts of knowledge, yet the intelligent manipulation of that knowledge is handled by a marker passing algorithm that can be described in terms of a simple grammar. FAUSTUS has a fixed memory which is rich in knowledge, but it is structured very carefully using a knowledge representation language called KODIAK [Wilensky 1986]. When activation passes from one concept to another, it must conform to a legal path "shape" specified by the grammar in the marker passing algorithm. When independent markers collide at a shared node, the resulting path of activated nodes provides useful inferences about the original input items. The idea of the intersection search is still there - it's just harder to generate false positives (bogus intersections).

The best way I can give you a feel for FAUSTUS is by looking at an example. The following example was manufactured for this talk and is undoubtedly all wrong as far as the details of KODIAK and Norvig's actual algorithm are concerned, but we'll settle for ballpark accuracy to get the main idea across.

Let's go back to our overworked text about the balloon and the light bulb. The first sentence was, "When the balloon touched the light bulb, it broke." We have a reference to a light bulb, a reference to a balloon, and physical contact between the two of them. That's explicit in the sentence. We also know something broke, but the pronoun leaves us up in the air as to exactly what broke. It could have been the light bulb or it could have been the balloon. We would like to be able to disambiguate the pronoun and infer a plausible causal relationship between the two events described. Figure 5 shows us what a meaning representation for the input sentence might look like before any inferences are made.

[insert figure 5 about here]

Now let's look at some knowledge we should have available to us. We have knowledge about breaking which tells us all the different ways things can break. For example, we can understand that one way things break is by exploding. An exploding event is a further specification or "concretion" of a breaking event, and this further specification is only valid under certain circumstances. Using KODIAK, we can create inheritance hierarchies which encode structured inheritance via role-play links. As we will see, this notion of structured inheritance will help us make some important inferences about what broke and exactly what the breaking event describes.

[insert figure 6 about here]

We have a hierarchy of entailed event concepts going from breaking down to exploding, with role-play links telling us how these structures are inherited. These hierarchies bottom out with very specific event descriptions: specific, for example, at the level of a balloon exploding (see figure 6). And we understand that there's a constraint on the balloon exploding event that the object of any such event must be a balloon. This is not a constraint available to us at the higher levels, where we may only be constrained by the specification of an inflatable object, or even more generally, a physical object.

A hierarchy with these richly constrained specifications allows us to generate concretion inferences which help us see beyond the explicit meanings available to us from the source text. For example, if we are told that a balloon broke, we should be able to infer the constraints operating at low levels of greater specificity in order to understand that if the object of a breaking event was a balloon, then it may be safe to assume that the balloon exploded.

Concretion inferences are one of the inference types handled by FAUSTUS, but the simple inheritance mechanism described above cannot resolve complicated ambiguities of the type present when we have to understand what it was that broke in the first place. In our original text, we have to decide between a balloon breaking or a light bulb breaking. It is nice to know that the balloon would break by exploding, whereas the light bulb would break by shattering (see figure 7), but we still have to decide which object we think we're dealing with.

[insert figure 7 about here]

If we really want to resolve the reference, we have to drag in more knowledge. So let's assume we have knowledge about balloons (see figure 8).

[insert figure 8 about here]

This is somewhat reminiscent of the balloon script we discussed earlier. We understand that one of the things that can happen to an inflated balloon is that it might come into contact with a hot object, in which case we can make a pretty fair prediction about a causal relationship with a balloon exploding event. The preconditions for this balloon exploding event can be obtained from the light bulb if we understand that a light bulb can be a hot light bulb, and that hot light bulbs are further specifications under turned-on light bulbs. With appropriate inheritance inferences (including the fact that a touching event is a further specification for physical contact, and the fact that an inflated balloon is a further specification for a balloon), we might manage to fill out a casual chain if all the pieces are available to us in memory and the paths of relevant inference are recognized by the marker passing grammar.

As this example shows, FAUSTUS attempts to marry extensive knowledge access to a homogeneous control structure realized in terms of marker passing. The approach represents an appealing synthesis of two seemingly contradictory directions: the weak methods of homogeneous control and the strong methods associated with large amounts of knowledge. However, it is difficult to say what happened to the strong methods associated with traditional knowledge structures when we encoded our knowledge base in KODIAK. Can a marker passing algorithm achieve the computational power of a script applier mechanism? Can generic concepts be instantiated and utilized by multiple referents without getting confused? What if our story references two balloons and we have to keep distinct concretions straight? These are questions about the possible limits of marker passing algorithms. The homogeneous control is great, but is it powerful enough for our needs? These are questions we need to answer about marker passing as a weak method for inference generation.

4 Syntax and Semantics

We've been talking a lot about inference generation, but it would be a mistake to assume that's all there is to knowledge-based natural language processing. In fact, homogeneous control for inferences really goes hand in hand with homogeneous control for other problems. For example, we are also seeing a trend toward homogeneous control for the integration of syntax and semantics, a problem which is very important for models of sentence analysis. Let's see how some people have worked to bring homogeneous control back down to the level of sentence analysis.

What do you usually see when you look at a textbook on AI with a section devoted to natural language processing? There's a good chance you'll see a flow of control diagram that looks something like this (see figure 9).

[insert figure 9 about here]

Here we see that the problem of sentence analysis has been divided into specific modules. We have syntactic knowledge - knowledge about grammar - that is important in analyzing the structure of a sentence. We also have semantic knowledge, which is where concept frames are defined, and various constraints operate to control the slot fillers for those frames. And we often see a reference to pragmatic knowledge, which is where all the common sense reasoning needed for inference generation resides. Pragmatics is also where knowledge about discourse is stored. Generally speaking, pragmatic knowledge is defined to be anything we need which wasn't already covered by syntax and semantics.

The flow of control that we see here is serial control. This is a nice modular idea about language analysis which lays out the pieces clearly and simply. Unfortunately, systems built along these lines just don't work very well. Serial control is used for some database interfaces, but it doesn't work for continuous narrative text at all.

To see why not, let's look at a couple of sentences (see figure 10).

[insert figure 10 about here]

The sentences I'm interested in are, "John took her flowers" and "A stranger took her money." These two sentences are syntactically identical, and they are syntactically ambiguous as well. "Her flowers" could be a single noun phrase, or it could be an indirect object followed by a direct object. Similarly, "her money" could be a single noun phrase, or it could be an indirect object followed by a direct object.

When Mary is in the hospital, we understand, without effort or conscious thought, that John brought flowers to Mary. The sentence contains an indirect object and a direct object. But when Mary is in Central Park, we see a single noun phrase operating as a direct object. Somehow we fail to consider the absurd possibilities of John taking flowers away from Mary in the hospital, or even sillier, the possibility that a stranger could walk up to Mary in Central Park and hand her money.

Apart from the syntactic ambiguities confronting us, we also have a lexical ambiguity associated with the verb "to take." In the hospital this verb means "to bring," while in Central Park we understand it to mean "to take away." This is a strictly semantic ambiguity which forces us to choose between competing word senses.

So we have two interesting ambiguities operating here. We have a syntactic ambiguity that needs to be resolved, and we have semantic ambiguity associated with multiple word senses. Both ambiguities must be resolved in order to arrive at appropriate interpretations for the sentences.

How do we do it? Well, first we note that there are useful relationships between syntax and semantics. When "take" is used to mean "bring," it predicts a different set of syntactic constituents than when "take" is used to mean "take away." When you take something away from someone, you can't have an indirect object. This means that a resolution of the semantic ambiguity will automatically take care of the syntactic ambiguity as a natural side-effect. Once we know what the verb means, we'll know how to parse the sentence syntactically. We'll return to the problem of knowing what the verb means in a minute.

In the meantime, notice that we're already in trouble using our serial architecture. This architecture assumes that all the syntactic decisions are made before we even look at the semantics of the sentence. The dependency is running the wrong way. If we stick with this architecture, we'll have to allow the syntax module to operate nondeterministically, handing multiple parse trees over to semantics in the hope that semantics can decide which one is appropriate.

This is, in fact, exactly what a lot of language processing systems do. In the "syntax-first" tradition, whole sentences are analyzed syntactically, and multiple parse trees are passed on for further analysis, making the job of semantic analysis a job of sorting through all the parse trees.

When sentences contain prepositional phrases, reduced relative clauses, and other sources of rich syntactic ambiguity, the number of syntactic parse trees available to us can easily run into the hundreds.

Most researchers in knowledge-based natural language processing reject the syntax-first approach to sentence analysis and strive to integrate syntax and semantics in a more natural and effective manner. But once we open the door to integrated models of sentence analysis, we must necessarily ask whether the problem is restricted only to syntax and semantics. After all, just how do we decide what word sense for "took" is the appropriate one?

It seems that the answer to this question must be obtained by using a lot of knowledge about the world. Although you may not have thought about it, you make an inference when you hear "Mary was in the hospital." Probably, Mary was a patient in the hospital (note that this could be wrong). It follows that Mary was probably sick or injured. And there's a tradition in our culture about people who are sick or injured. Friends and relatives usually send something to cheer up the invalid: cards and flowers are traditional items. All of this is useful in disambiguating the proper word sense in "John took her flowers." Given the strong context surrounding the sentence, we might reasonably expect to be dealing with a bringing event as soon as we hear "John took ..."

On the other hand, we also have knowledge about Central Park. We all have a strong association between Central Park and muggers, we know what a mugging is, what the goals of a mugger are, and we know that pedestrians in Central Park are at risk. All of this is available to most adult Americans because it's a part of our shared culture. And this is the knowledge that helps us to understand the appropriate word sense for the verb when we hear "A stranger took ..." in the context of pedestrians and Central Park.

If we define pragmatic knowledge to be the basis for inference generation, then we have to integrate not just semantics with syntax, but semantics and pragmatics with syntax as well. For this reason, many people believe that the line between semantics and pragmatics is not well-motivated: there is no good basis for distinguishing semantic knowledge from pragmatic knowledge if you are going to work within an integrated framework for sentence analysis.

People who are interested in this integration problem are interested in ideas for control. How are we going to integrate the top-down processes which are knowledge-based with low-level bottom-up processes which are not knowledge based? Although there are many answers to this question based on co-routines and message passing, it has been difficult to find solutions that are truly elegant and readily adaptable if your grammar changes or your theory of semantics begins to shift.

However, two interesting approaches to this problem have surfaced very recently, and I'd like to give you a rough feeling for those solutions. I am not convinced that anyone has a good solution to the pragmatic context effects we've been looking at in figure 10, but we can at least see progress at the level of syntax and semantics with hopeful hand waving aimed at pragmatic interactions.

In the first case, structured inheritance is being pushed as a key mechanism for integrated sentence analysis. This approach argues that the key to the problem lies in the correct design and organization of our knowledge base. For example, a selling event can be characterized in terms of two transfer events, where the object of one transfer is money and the object of the other transfer is merchandise. The sources and recipients for these two transfer events constrain one another by exchanging roles, and at a very high level of abstraction, each of these transfer events are instances of some very vague event which corresponds to the primitive ATRANS in Conceptual Dependency. Figure 11 shows how all of this knowledge about selling might be represented using KODIAK.

[insert figure 11 about here]

In KODIAK diagrams we use a bit of shorthand which is important to understand. Whenever you see a named link like the actor link in figure 12, that's actually a shorthand notation for structured inheritance via a role-play link. It's very cumbersome to work with the fully expanded notation all the time, so the shorthand notation is useful, but we must remember that this shorthand implies a structured inheritance that is not explicit in the diagram.

[insert figure 12 about here]

What we're trying to do here is create a very systematic and highly constrained style of knowledge representation through which we inherit a lot of implicit structure as needed. Let's try to look at some examples of this in action.

Selling is interesting because it's two transactions, and both of those transactions are transfers. We have some very high level of generality, a transfer of an object from one person to another, or from one entity to another. And in one case, the transfer is a merchandise transfer, so have an object of barter being moved from one person to another. In the other case, moving in the opposite direction is a transfer of tender: money is changing hands. If we're very careful with our representation, we can understand how these two transfers relate to one another. They are not isolated transfers. Rather, they are connected through a series of links that identify specific roles, such as customer, merchant, merchandise, tender. Whenever there's a selling event, we implicitly know that four roles must be present, whether we can instantiate them with referents or not.

While this network is designed to represent semantic information, the idea of structured inheritance networks has been applied to traditionally linguistic (syntactic) knowledge as well

[Jacobs 1987a]. It is possible to take knowledge about grammar, the rules for recognizing legitimate sentence structure, and encode that knowledge in a KODIAK network utilizing structured inheritance. Once this is done, we have our linguistic knowledge together with the semantic knowledge within a single representational framework (see figure 13).

[insert figure 13 about here]

Concretion mechanisms (or any other marker passing algorithm) that worked for inference generation can now be applied to syntactic structures as well since the underlying data structures are indistinguishable. Whether all such mechanisms generalize to useful applications is another question, but at least we are now in a position to ask.

Although we are concentrating here on techniques for sentence analysis, it is interesting to note that the integrated KODIAK structures we've been discussing are used for both sentence analysis and sentence generation [Jacobs 1987b].

Although Jacobs is probably the first researcher to investigate highly integrated methods for syntactic/semantic processing from the two perspectives of analysis and generation, he was not the first to work with a uniform representational framework for sentence analysis. The earlier Word Expert Parsing effort [Small 1980] deserves to be mentioned along with related work on lexical access [Cottrell and Small 1983] which focused on the problem of word sense ambiguity.

A very different approach to the problem of integrating syntax and semantics can be found in an effort which was strongly influenced by Cottrell and Small's earlier work. Waltz and Pollack [Waltz and Pollack 1985] picked up where Cottrell and Small left off, and tried to generalize connectionist techniques into higher levels of sentence analysis. While we have seen a lot of exciting work by connectionists on sentence analysis within the last year or two (see for example, McClelland and Kawamoto 1986), I've chosen to talk about Waltz and Pollack because the techniques they use are much more accessible to an AI audience without an introductory tutorial on connectionism.

Waltz and Pollack work with large, knowledge-rich networks in their system, but these networks are not as carefully structured as the KODIAK networks we saw before. Indeed, one of the weaknesses of this system is it's lack of inheritance in any form. There are no theoretical claims about knowledge representation here either: one could invent a node for any sort of frame with additional nodes for any kind of role or slot constraint imaginable.

The key idea here is spreading activation and network relaxation. But now the activation is analog activation which means that nodes are given numerical values to indicate how much activation is present at any given time. Relaxation is the process of systematically adjusting activation levels within the network until the network assumes a stable state. A stronger connectionist flavor is obtained by the use of lateral inhibition to expedite the stabilization of competing nodes where activation levels are expected to be mutually exclusive. If we appear to have walked off some sort of cliff in terms of your familiarity with these terms, that's probably because this is a numerical algorithm and not the sort of thing we normally associate with "mainstream" symbolic AI.

[insert figure 14 about here]

Consider, for example, an eating node, which has arcs leading out to role nodes that represent things like agents and objects (see figure 14). When we understand the sentence "Mary ate spaghetti with Sue," we want to see the network stabilize with a high level of activation on this eating node as well as the appropriate slot filling nodes. It is important to settle on a high level of activation for the co-agent node lest we interpret Sue to be a co-object (like meatballs) or instrument (like fork) for the eating event. If all goes well, semantic constraints within the network will push the relaxation process in the right direction, and inappropriate pathways in the network will die off for lack of sufficient activation.

If ever there was an algorithm to illustrate homogeneous control, numerical relaxation must be it. This idea can be applied to networks of nodes representing anything you want. We can have different nodes for different word senses, other nodes for semantic features, and even nodes for traditional syntactic constituents. Plug in a grammar by wiring the nodes correctly, and you can produce syntactic parse trees as a side-effect of network relaxation (see figure 15).

[insert figure 15 about here]

Within this framework we integrate semantic constraints and syntactic constraints in a massively parallel architecture that can readily compute a global assessment of the situation after each word of the sentence is received. Preferred word senses and syntactic preferences may shift around as we move through the sentence, making it possible to run interesting experiments by taking "snapshots" of the network as we move through a sentence. Activation levels from a syntactic constituent may inhibit or support a specific semantic interpretation, and semantic preferences can flow back toward the nodes deciding about syntax.

This provides us with a very nice framework for investigating a lot of problems, and in particular, garden path processing phenomena are especially well suited for analog spreading activation models. Of course, all of the problems we have with marker passing algorithms apply here as well: e.g. what happens if two different referents activate the same sections of the network? In fact, the interference effects associated with analog activation are even worse than with marker passing algorithms because we have to make sure that nodes "die out" within a reasonable period of time by tweaking the numeric algorithm. In a marker passing framework, a node can be told to die after a fixed number of words have been parsed or after a specific marker like a clause boundary is encountered. In the symbolic paradigm it is at least easier to understand why a node is turned on or off. In the analog paradigm, the status of each node is dependent on the status of every other node in the network, making the whole business rather inscrutable.

Now that we've seen how syntax and semantics might be intertwined under homogeneous control, let's return to the issue of pragmatics and how processes of inference might be interleaved with processes of sentence analysis. As I said earlier, I don't think a lot of progress has been made in this area. Waltz and Pollack have designated a subset of their nodes as "context nodes," but it is difficult to evaluate the utility of that idea in the absence of a systematic methodology for building large, massively parallel networks. Probably the best I can do is show you some more places where "high-level" knowledge must be allowed to influence "low-level" decisions about syntax. One of the places where this appears to happen involves analogies and the role of analogical thinking in natural language.

5 Analogical Reasoning and Language

"Her hair was like lamb's wool, her teeth were like pearls."

We're supposed to understand from this that her hair was soft and her teeth were white. We're not supposed to conclude that her hair was white and her teeth were hard. One discovers that the mapping of a sentence onto appropriate analogical features is not such a simple business. Perhaps her hair was smelly and her teeth were very round?

We heard a survey talk earlier today by Deidre Gentner on analogy. Analogical reasoning is a major problem in natural language communication, and we don't have to reach for poetry to find instances of it. In fact, it's much more common than you might imagine. Sometimes we see it explicitly, in the example above. The word "like" warns us that we may be talking about an analogy and we'd better get the mapping right. But analogies can also operate more subtly.

For example, idioms often rely on analogies of one sort or another. I can pick up an article in the newspaper and read about a conflict in the Middle East: "Despite the fact that the two factions had been fighting for 20 years, they finally agreed to bury the hatchet." This is a standard idiom. Everyone understands what is meant by it. Or we can go back to Mary in the hospital. Maybe after John took her flowers, she took a turn for the worse and kicked the bucket. Another idiom. In fact, there were two idioms in there. Nobody I know can take a turn for the inferior.

For a long time, no one in AI had much to say about idioms. They were just conventionalized and fossilized expressions in the language - a part of the phrasal lexicon that had to be learned case by case. But if you look at it with analogy in mind, there are some very interesting phenomena associated with idioms. To be precise, there appear to be some rules that govern the syntactic flexibility of idioms, and those rules are based on analogical reasoning processes.

First, we must understand that some idioms are more fossilized than others. The burying of the hatchet can be passivized: "After the peace talks, the hatchet was buried." The kicking of the bucket cannot be passivized: "After a long illness, the bucket was kicked by Mary." That's just not an option. One of these idioms can tolerate a syntactic transformation while the other can't.

In a recent Ph.D. thesis we find a claim about this [Zernik 1987]. The key question is whether or not a given idiom can be explained via analogical reasoning. If an idiom can be explained, then it will be syntactically flexible. If it can't be explained, then it will be brittle.

Let's look at this in a little more detail.

In the case of the hatchet, we have associations and we have knowledge. You always have to have knowledge in order to have an analogy. And the knowledge that's relevant here is knowledge about war. One can imagine a war script, where we have stereotypic events. You have some initial conflict, you gather your troops, you attack, you defend, you win, lose, draw, you establish an agreement, and you bring your troops home. Somehow, we have to get from burying the hatchet, which is a very specific literal event, to the withdrawal of armed troops. If we can make that connection, then the hatchet operates as an instrument of aggression (just as the armed troops are a symbol of aggression), and burying the hatchet translates into a deliberate disarmament, a halt to aggression.

How do you make those connections? This is a very difficult problem for knowledge representation and memory organization. We could call it a concretion problem, but that doesn't exactly solve anything. Is there an abstract event that dominates both troop withdrawals and hatchet burials in some massive inheritance hierarchy? If we go up the abstraction hierarchy too far, all events will map to all other events (because they're all dominated by some very general event node way up at the top).

Concretion by itself is probably too powerful a mechanism in the sense that it could be used to make sense out of idioms no one ever heard of. If burying a hatchet is a further specification of weapon burial, then burying a rifle should be recognized just as easily as burying the hatchet. Somehow we lost track of the fact that one of these is an idiom and the other is not. What distinguishes the one from the other is an instance (real or plausibly constructable) where someone actually buried a hatchet following a conflict. Perhaps we all remember a story about the pilgrims and the Indians from our 4th grade history lessons. It's at least conceivable that an Indian might have buried a hatchet in a war ritual. To bury a rifle is to impose an event from a ritually rich culture on an object from a culture largely lacking in symbolic rituals. The mismatch arouses cognitive inconsistency and seems disturbing.

Ignoring the very difficult problems associated with analogical reasoning, we can hypothesize that some such processes take place. Or at least they take place for the idioms that can be explained. If we had to explain "burying the hatchet" to a child, we would probably describe a scenario where a hatchet got buried to symbolize the end of physical aggressions. But what would you do if someone asked you to explain "kicking the bucket?" Most people explain this one by saying it's just an expression (don't bother me kid). There is no analogical mapping that gives us a plausible explanation for why death is associated with kicking a bucket. Most of us do not know of any such explanations and can't construct a plausible one even if we try.

So why should any of this matter to a syntactic transformation? The fact that some idioms are syntactically flexible while others are not suggests that the processes associated with the two types of idioms are very different. An explainable idiom is understood at a deep conceptual level ... the idiom maps into a conceptual structure retrieved by analogical reasoning. An inexplicable idiom is understood (she kicked the bucket $\Longrightarrow$ she died) but not explained by analogical mappings.

When an explanation is available, all of the language processing power available for the targeted conceptual structures can be applied. The explanatory concept underneath the idiom can be expressed using a variety of syntactic structures, and this makes the idiom receptive to syntactic transformations. When no explanation is available, there is no underlying concept associated with the idiom, and so there is no language processing capability that applies. Brittle idioms lack the conceptual scaffolding required to loosen them up.

Before we leave the topic of analogical reasoning, I want to give you some more examples of its utility for natural language. One way that analogical reasoning creeps in is via metaphor. Metaphors are abundant in natural language, and so pervasive we don't even notice them most of the time. For example, it is common to assume that technical literature is characterized by very dry and literal language. If there is one place where metaphors might not intrude, it must be when people discuss technical or scientific concepts.

Surprisingly, technical descriptions are often very rich in metaphors. Consider, for example, the language we commonly use when talking about computers:

You can *get into* the editor by...

I *ran* it *through* spell to...

The editor *died* when...

If you have a language processing system that assumes only living things can die, you're going to have a lot of trouble with a sentence like "The editor died on me." [Wilensky, Arens and Chin 1984]

Oliver North has given us a beautiful example of how intimately interdependent language and analogical reasoning can be. If you were listening to the Congressional hearings last week you heard Col. North explain a misunderstanding he had about the term "delete" in the context of electronic mail. He thought that when you pushed the delete button, the mail really went away.

I suspect that this faulty interpretation of deletion was the direct result of an analogical mapping to a bad analogy. Given the rest of his testimony before the Congressional hearing, it seems quite likely that Col. North mapped the delete command in his mail system to the on button of a paper shredding machine. When you turn on the shredding machine, things really do go away. Unfortunately, shredding machines are not very good models for what happens to electronic mail. If Col. North had ever worked with icon-infested software of the sort found on personal computers, he might have mapped the delete command to a wastepaper basket, and been more concerned about the security of his deleted documents for the same reason that one should worry about wastepaper baskets.

I do not mean to disparage Col. North or his memory organization. This kind of misunderstanding happens to all of us and it's especially dangerous when a word appears to be so simple. How do people usually explain something like a delete command? When you say delete, the message will go away. When you delete a message you throw it out. Deleting a message destroys the message. None of these explanations are quite correct but how many of us really want technically correct explanations? Natural language communications are generally very effective in trading off accuracy for brevity. But every so often the trade-off slips up and mistakes result. What's amazing is how we all get by as well as we do.

6 Episodic and Semantic Memory

Let me close on a topic that is in keeping with our theme of homogeneity. In addition to homogeneous control, we can talk about homogeneous memory. There's some very interesting work which I think is just beginning to get off the ground. The one example that I'll draw from in order to illustrate what I'm talking about is some recent work done at Yale [Riesbeck and Martin 1986].

Traditionally, people who talk about memory make a distinction between semantic memory and episodic memory. To understand this distinction, let's think about how we might go about answering a simple question. Suppose I ask you, "Does a penguin have skin?" If you have a semantic memory available to you that involves penguins, you will understand that a penguin is a type of bird, and as a bird, it has specific features, one of which is skin. If you have any kind of retrieval algorithm available for answering questions, you will traverse links of this sort in order to confirm that penguins do indeed have skin.

Now suppose I ask a very similar question. What about a chicken? "Does a chicken have skin?" Now, if you have semantic memory, you're going to answer the question much the same way you answered it for penguins. You won't have associations available to you about Antarctica, but you'll find chickens, you'll find birds, you'll find features for birds, and you'll find skin. Just like before. This is the semantic view of memory.

However, a number of people believe something else goes on, that perhaps semantic memory can sometimes be short-circuited by something much scruffier called episodic memory. Episodic memory has to do with personal first-hand experience with the world. For example, dinner last night is a good example of episodic knowledge. If dinner last night happened to be fried chicken and you really like the skin on fried chicken, you might have a much faster path for answering the question about chicken skin than the one available through semantic memory (see figure 16).

[insert figure 16 about here]

Traditionally, semantic knowledge and episodic knowledge have always been thought to be in competition with one another: these are two distinct views of memory and there really isn't room in this world for both of them to coexist peaceably [Tulving 1972].

But very recently we've begun to see some work which seems to blur the semantic/episodic barrier and cross lines between the two without any trouble at all. We've already seen some of this with FAUSTUS. What sort of a node is the node that represents balloons exploding? An exploding balloon sounds pretty episodic. Yet two steps up the hierarchy we'll see general nodes for explosions and breaking events. Nodes like that are commonly found in semantic networks. If we examine the memory structures engineered for FAUSTUS, it seems that the task of inference generation needs both types of memory and would be badly impaired if forced to function without one or the other.

Now let's get back to Riesbeck and Martin to see how the semantic/episodic issue relates to sentence analysis. Before describing their system, DMAP (Direct Memory Access Parsing), Riesbeck makes an interesting claim about language analysis at the level of sentence comprehension. He points out that there are really two distinct views about what it means to analyze a sentence. In one perspective, we think of a sentence as mapping into existing concepts in memory. That is, you really only understand this sentence because you have knowledge in memory which allowed you to make sense out of it. Then when you understand the sentence, the very act of understanding the sentence operates to reinforce or modify existing structures in memory. This view of sentence analysis might not sound terribly controversial, until you realize that virtually every sentence analyzer ever implemented operates under different premises.

In most models of sentence analysis, sentences do not map directly into memory. They create meaning representations, and these meaning representations may be influenced by some form of memory, but the act of sentence analysis rarely has any side-effects that alter memory as the target meaning representation is being produced. The processes that analyze a sentence are normally segregated from the processes that alter memory (if indeed, any process is capable of altering memory).

Riesbeck characterizes the traditional framework as the "build-and-store" approach to sentence analysis. He calls the non-traditional framework the "recognize-and-record" style of sentence analysis. He then goes on to argue that it would be much to our advantage to investigate recognize-and-record models of parsing as a wholly new style of parsing that lends itself more naturally to a truly memory intensive view of language.

In fairness, we should point out that the Waltz and Pollack parser falls somewhere in between build-and-store and recognize-and-record. Their analyzer produces a pattern of activation over its entire memory. Indeed, it may be very difficult to interpret this pattern of activation should anyone ever need to know what a particular sentence means. So Pollack and Waltz are certainly not consistent with the build-and-store paradigm. On the other hand, the changes made to memory as a result of sentence analysis are completely transient and wiped out each time a new sentence is processed. So this is not exactly consistent with the recognize-and-record idea either. Yet the connectionist enterprise in general is clearly operating within the recognize-and-record paradigm if we look at the learning algorithms that adjust weights and modify the network each time a new sentence is processed. The radical view that Riesbeck advocates is really only radical within symbolic AI circles. Connectionists would feel quite at home with it.

To see how Riesbeck and Martin try to realize a recognize-and-record model using symbolic techniques, let's look at one of their example sentences. Here is a picture of DMAP's memory (see figure 17).

[insert figure 17 about here]

DMAP has some knowledge about newspaper articles taken from newspapers. The sentence we are now trying to understand is, "Interest rates will rise as an inevitable consequence of the monetary explosion." This is a quote from Milton Friedman in the New York Times. Figure 17 shows us the portion of DMAP's memory which is important for understanding "(Milton Friedman says) interest rates will rise ..."

At the highest level of memory, we can characterize this sentence as a transfer of information. Somebody said something. This is a highly abstract characterization of the input sentence. As we move down to a more specific representation, we further understand the sentence to be an opinion by an economist. Even more specifically, a prediction by an economist. And more specifically again, a prediction by Milton Friedman about interest rates.

Looking at figure 17, we can see an inheritance hierarchy that gives us all the further specifications needed to represent the input at various levels of abstraction. If we start at the top node for a communication event, filling in the details becomes something like a concretion problem. Of course, memory will only look like this if DMAP has already seen other stories about Milton Friedman making predictions about interest rates. Given such knowledge, the act of mapping our new input sentence into memory becomes an act of recognition: I see now ... this is another interest rate prediction by Milton Friedman. DMAP shows how a sentence analyzer can work with memory in order to situate the content of a sentence within an existing framework for memory. The algorithm is a marker passing algorithm, and DMAP shows us what sentence analysis might look like within a memory-rich recognize-and-record paradigm.

Let's take one more look at the nodes in this tree structure (see figure 17). Although the root node for a communication event looks very generic and therefore semantic, nodes further down the tree structure look more and more episodic. We have a node for all the names we know with the first name Milton. We have a node for economic predictions by Milton Friedman. This is completely episodic.

At some point, we've crossed the line and moved from nice, clean, semantic knowledge down to scruffy, first-hand experience knowledge of Milton Friedman and what he's said in the past. In fact, the marker passing algorithm in DMAP was designed with two kinds of memory organization in mind: abstraction hierarchies and packaging hierarchies [Schank 1982]. The abstraction hierarchy is the traditional is-a hierarchy we see in semantic networks, and the packaging hierarchy handles stereotypic chronologies of the sort we first saw with scripts - this is clearly episodic knowledge.

So an interesting line gets crossed in DMAP, and there are important implications when you cross that line. One of the implications has to do with knowledge acquisition. If you are willing to cross that line and benefit from the advantages associated with it, then you necessarily have to worry about knowledge acquisition. Because every time you understand a sentence, you should add another instance of something to your knowledge framework. The tenth time you read about Milton Friedman predicting interest rates will rise, you should feel that the concept is somehow more familiar than it was the second time around. You are automatically in the learning business at that point. Earlier work on generalization and dynamic memory organization come to mind [Lebowitz 1983]. But this is a not a standard perspective on sentence analysis. Most researchers in natural language processing and even knowledge-based natural language processing would not claim to be working on learning or knowledge acquisition. So this is a really a radical view of language being promoted here.

7 Conclusions

That brings us to our wrap-up. I've tried to point out some trends over the last 15 years. It is possible to associate the trends with roughly 5-year cycles starting in 1972.

The first cycle (1972-77) was characterized by a preoccupation with strong methods addressing specific knowledge structures and processes of inference associated with specific knowledge structures. Ph.D. theses by Charniak and Rieger motivated much of this work, and Schank organized a large research group at Yale to identify knowledge structures for natural language processing.

The second cycle (1977-82) was characterized by a gradual appreciation for the implications of language processing based on strong methods alone. Dyer's thesis gave us a taste of the price we would have to pay in terms of system complexity if the strong methods continued to propagate without other kinds of processing techniques. At the same time, powerful ideas based on the earlier impetus toward strong methods were being pushed hard and refined in a number of computer implementations. Jaime Carbonell, Richard Cullingford, Gerald DeJong, Michael Dyer Richard Granger, Janet Kolodner, James Meehan, Mallory Selfridge, Robert Wilensky and I, all finished theses at Yale during this period. The pendulum was poised to swing back from there.

The third cycle (1982-87) fueled a renewed interest in weak methods - techniques for homogeneous inference generation, homogeneous memory organization, and broad processing techniques of great generality. Marker passing algorithms enjoyed a lot of attention during this period and progress by connectionists was greeted with cautious enthusiasm. Spreading activation became a common theme in a lot of the original research of this period. James Hendler, Graeme Hirst, Paul Jacobs, Peter Norvig, and Jordan Pollack, all completed theses consistent with the Zeitgeist of this cycle. Work by Gary Cottrell and Steve Small received attention for earlier work which surfaced "before its time."

So where are we going in the next five years? It's always safer to wait for 20-20 hindsight, but I'm willing to stick my neck out and imagine a future that would at least would not surprise me.

- I expect to see a push toward knowledge acquisition as an active concern in knowledge-based natural language.
- The symbolic community will grapple with the questions raised by connectionist research: What are the essential issues in the symbolic/subsymbolic paradigm struggle? Should

we all see the light and become connectionists? Should the connectionists see the light and forsake connectionism? Given the unlikelihood of those two scenarios, how will the two communities come to view each other and the relationship between their distinctive research paradigms?

- Somewhere in the midst of all this, theoretical progress might be made on the episodic/semantic distinction. More and more people will find it convenient to acknowledge the utility of both memory types and design algorithms that move freely between them. This will be viewed either in terms of an integration of two distinct memory types, or a demonstration that the original distinction cannot be supported by computational models (it was a bad idea in the first place).

- Finally, we may see some serious efforts aimed at evaluating our models and understanding the qualitatively different contributions that are being made by different research styles. The neat/scruffy dichotomy may give way to some other, more timely wedge, as more and more people find it difficult to pigeon-hole themselves as card-carrying neats or free-spirited scruffies. Those who never liked this distinction in the first place will hold a workshop and burn all reprints that contain the keywords "neat" or "scruffy."

In closing I'll leave you with two of my favorite quotes. The first one is by Thomas Edison. Thomas Edison was born too early to be an AI person, but I think he would have been a good one if persistence counts for anything. He had a lot of trouble finding the right filament for the light bulb, and he tried a lot of filaments before he found a workable one. Whenever I see the following quote I like to mentally transport Edison into 1987 and place him in an NSF office where he's trying to convince a program manager to fund his research. Exasperated and impatient with the obvious difficulty of his situation, he says:

"I've tried everything. I have not failed. I've just found 10,000 ways that won't work."

I think anyone who's been in AI for more than ten years can probably relate to that scenario, but this is a rather pessimistic perspective on the state of the art, so I don't really want to leave you on that note. It makes the whole business sound like a simple brute force search, and I think we're all at least a little smarter than that.

Here's a happier observation from Francis Bacon which seems closer to the true spirit of AI:

"Truth emerges more readily from error than from confusion."

8 Acknowledgements

This research was supported by DARPA contract #N00014-87-K-0238.

9 References

Abelson, R. (1981). "Constraint, Construal, and Cognitive Science". *Proceedings of the Third Annual Conference of the Cognitive Science Society.* Berkeley, CA.

Alker, H.R., Jr., Lehnert, W.G., and Schneider, D.K. (1985). "Two Reinterpretations of Toynbee's JESUS: Explorations in Computational Hermeneutics". in *Artificial Intelligence and Text-Understanding: Plot Units and Summarization Procedures, Quaderni di Ricerca Linguistica*, edited by Graziella Tonfoni.

Bobrow, D.G., and Winograd, T. (1977). "An Overview of KRL-0, a knowledge representation language". in *Cognitive Science*, vol. 1(1): 3-46.

Charniak, E. (1972). "Toward A Model of Children's Story Comprehension". Massachusetts

Institute of Technology Artificial Intelligence Laboratory, Cambridge, MA. AI TR-266.

Cottrell, G.W. and Small, S.L. (1983). "A Connectionist Scheme for Modeling Word Sense Disambiguation", in *Cognition and Brain Theory*, vol. 6, No. 1.

Cullingford, R. (1978). "Script Application: Computer Understanding of Newspaper Stories". Yale University, Dept. of Computer Science, Research Report #116. Dissertation.

Dreyfus H. (1987). "Artificial Intelligence: Where Are We?" *ABACUS*, Vol. 4, No. 3. (An article edited by Bobrow, D.G. and Hayes, P.J.). pp. 13.

Dyer, M. (1983a). *In-Depth Understanding.* MIT Press,, Cambridge, MA.

Dyer, M. (1983b). "The Role of Affect in Narratives". *Cognitive Science*, 7, 211-242.

Galambos, J., Abelson, R., and Black, J. (1986). *Knowledge Structures*, Lawrence Erlbaum Assoc., Hillsdale, NJ.

Geisel, T. (1950). *If I Ran the Zoo.* Random House, New York.

Jacobs, P. (1987a). "A Knowledge Framework for Natural Language Analysis". *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milano, Italy. pp. 675-678.

Jacobs, P. (1987b). "Knowledge-Intensive Natural Language Generation". in Artificial Intelligence, Vol. 33, No. 3.

Lebowitz, M. (1983). "Memory Based Parsing". Artificial Intelligence, 21, 4. pp. 363-404.

Lehnert, W.G. (1981). "Plot Units and Narrative Summarization". in *Cognitive Science*, Vol.5, No. 4.

Lehnert, W.G., Dyer, M., Johnson, P. Yang, C. and Harley, S. (1983). "BORIS - An

Experiment in In-Depth Understanding of Narratives". in *Artificial Intelligence*, Vol. 20, pp.15-62.

McClelland, J., and Kawamoto, A. (1986) "Mechanisms of Sentence Processing: Assigning Roles to Constituents". in *Parallel Distributed Processing: Explorations in the Microstructures of Cognition* - vol. 2 (eds: Rumelhart and McClelland). Bradford Books.

Norvig, P. (1987). "Unified Theory of Inference for Text Understanding". Dept. of Computer Science, University of California, Berkeley. Dissertation.

Quillian, M.R. (1968). "Semantic Memory." in Minsky (ed.), *Semantic Information processing*, MIT Press, Cambridge, MA.

Rieger, C. (1974). "Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances". Dept. of Computer Science, Stanford, Univ. Memo AIM-233, Stan-CS-74-419. Disseration.

Riesbeck, C. and Martin, C. (1986). "Direct Memory Access Parsing". in *Experience, Memory and Reasoning*, Riesbeck, C. and Kolodner, J. (eds.). Lawrence Erlbaum, Hillsdale, NJ.

Schank, R. (1982). *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*, Cambridge University Press.

Schank, R. (1975). *Conceptual Information Processing.* American Elsevier, New York.

Schank, R. and Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Assoc., Hillsdale, NJ.

Small, S. (1980). "Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding" Department of Computer Science, University of Maryland, TR-954.

Dissertation.

Tulving, E. (1972). "Episodic and Semantic Memory", in Tulving E, Donaldson (eds.) *Organization of Memory*, Academic Press, New York.

Waltz, D. and Pollack, J. (1985). "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation". in *Cognitive Science*, Vol 9, no. 1,

Wilensky, R. (1986). "Knowledge Representation - A Critique A Proposal." in Kolodner, J. and Riesbeck, C. (eds.) *Experience, Memory, and Reasoning*, Lawrence Erlbaum Assoc.

Wilensky, R. (1978). "Understanding Goal-Based Stories", Dept. of Computer Science, Yale, University. Research Report #140. Dissertation.

Wilensky, R., Arens, Y., and Chin, D. (1984). "Talking to UNIX in English: An Overview of UC." *Communications of the Association for Computing Machinery.*

Winograd T. (1987). "Natural Language: The Continuing Challenge". *AI Expert*, Vol. 2, No. 5, 1987. pp. 7-8.

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

Winograd T., and Flores, F. (1986). *Understanding Computers and Cognition*, Ablex Publishing Corp., Norwood, NJ.

Zernik, U. (1987). "Strategies in Language Acquisition: Learning Phrases from Examples in Context", UCLA-AI-87-1. Dissertation.

QUESTIONS AND ANSWERS

Q: I wonder if you might have seen the little note on USENET from Donald Normal about artificial intelligence as a science. Whether you have or not, let me ask the question. What, in your opinion, controls the development of this research from the point of view of both evidential support and falsification? I ask it because you didn't say anything about it.

A: Well, I think there's a lot of soul searching that goes on in AI on this point, particularly within the machine learning community. Language researchers are perhaps less preoccupied with such concerns because it is very hard to design convincing experiments for processes of this complexity. However, one good collection of psychological experiments inspired by the knowledge structuring work at Yale is [Galambos et al. 1986].

I think a big part of our enterprise can be reasonably characterized as trying to understand the problem before we can presume to find solutions. For example, Rieger thought the inference problem was primarily a control issue. Schank says it's primarily an issue about knowledge and memory organization.

I think we understand a good deal more about language now than we did 15 years ago, but whether we're learning what we learn by practicing a normal science is another issue. Personally speaking, I don't really care if we're practicing science as long as we can say we're learning something.

How about an easy question?

Q: I'll give you a technical question I have about the last point of your talk ... where you describe the recent work by Riesbeck as an effort combining episodic memory with semantic memory. You said that would create a problem for knowledge acquisition. It seems to me that if you could store the sentences you understand in the same representation that you are using to parse them, then that would be a big windfall for knowledge acquisition, because once you parse it, you have it available as part of your episodic memory for use later on. So the impression I get is just the opposite of what you said. Can you clarify that?

A: You have to careful about exactly what it is you think you should learn. If you're interested in psychological validity, there's a lot of evidence that people are very bad at remembering sentences verbatim in long-term recall or recognition. Even so, the content of those same sentences can be recalled. This suggests that our episodic memory structures operate with some system of knowledge representation that is not dependent on sentences per se.

When we say that DMAP can "understand" a sentence better if it's seen the sentence before, we should keep in mind that DMAP will also understand a paraphrase of the that sentence with equal advantage because the memory which facilitates understanding is based on a canonical form for meaning representation: all semantically invariant paraphrases are collapsed to into a single meaning representation. So DMAP can't be expected to learn anything about syntax or the processes needed to handle syntactic information as long as its memory can't record distinctions specific to syntax.

It is very difficult to say how the learning associated with episodic domain knowledge relates to the problem of learning how to analyze sentences. Going back to psychological validity, children acquire the basics of sentence analysis very early on. By the time a child enters school, she's basically working on vocabulary acquisition and an increasing tolerance for syntactic complexity — the hard part of language acquisition is over and what remains is a lot of expansion within existing structures. This suggests that the mechanisms associated with adult language processing are probably not very plastic or sensitive to specific sentences on a case by case basis. It might therefore make sense to separate the two types of learning as distinct and separable problems (as DMAP does). Of course, there are plenty of connectionists who would disagree with me about this.

Q: You spent some time talking about how one could use the same knowledge representation structures for representing the concept in the sentence and concepts of just verb and noun through grammatical terms, but I guess I missed something along the way. What power does that give you, what's the advantage of doing that?

A: Ah. Well, the idea is that we should get away from that one slide I showed you from Dyer's thesis, where the 22 different knowledge structures interact with one another in very arbitrary and idiosyncratic ways. If we could find knowledge representation techniques and memory organization techniques which allow us to bring in all kinds of different knowledge structures under the same representational umbrella, then we can develop algorithms that manipulate that information in a uniform manner. So it's a question of finding uniform processing theories as opposed to allowing the whole enterprise to break down into 1001 interacting experts who each speak different languages and talk about different things.

I should also point out that I'm only trying to identify some trends in our research. Time will tell whether or not this trend is justified. Maybe reality will ultimately reveal herself to be 1001 different experts and we'll just have to develop appropriate techniques for dealing with that kind of complexity.

Q: So in the case of Waltz and Pollack, we've really got sentences being parsed using only spreading activation? Some form of connectionism?

A: In the case of Waltz and Pollack, that's exactly what we've got. In the case of Jacobs who was working with KODIAK, we see another form of spreading activation called marker passing which operates a lot like relaxation except it's just not numerical relaxation. In both the numeric and non-numeric approaches, a simple algorithm is iteratively applied to nodes in the network until a stable state is reached. A lot of people are playing around with marker passing these days, including Charniak.

Q: And do those parsing algorithms duplicate the same phenomena that something like the Marcus parser does ... garden path phenomena?

A: Pollack and Waltz were very interested in garden path sentence processing and they have examples which simulate effects exhibited by human subjects.

Q: Could you speak briefly about the current interaction between psycholinguistics and computer science in language understanding, because it seems like some of these models come from insights from psycholinguistics, but you didn't mention that.

A: I think if you concentrate on the knowledge-based aspects of language processing, you find influence coming in from a number of places. For example, the Zernik work on frozen idioms and analogical mappings was, I suspect, heavily influenced or at least inspired by the work of George Lakoff.

Much of psycholinguistics, however, restricts its domain of inquiry to syntactic phenomena without appropriate concern for interactions between syntax and other knowledge structures. To the extent that this is true, many of the results we see from those experiments are not very illuminating for people working on knowledge-based natural language. Indeed, most of us argue rather vehemently against the segregation of syntactic processing.

Q: No, but the psycholinguists do experiment on memory, and they're interested in memory, they're interested in semantic memory, they're interested in cross-cultural effects of understanding. I was just wondering if there are any active relationships between these bodies of research.

A: There are scattered instances of influence. For example, Eugene Charniak was strongly influenced by the experiments of David Swinney in the late 70's. Experiments by Robert Milne are important for people working on lexical access. I'm not sure how much there is in terms of active collaboration, but it is always important to keep the channels of communication open.

Q: I've noticed that the entire description stayed within the verbal domain, and I'm wondering if that reflects a supposition about how people really think. Or is that just a starting point which we might have to move away from at some later time?

A: What do you mean by "verbal" domain?

Q: Well, for instance, when you said, "Does a penguin have skin?" I immediately saw a picture of a penguin. As a matter of fact, it was superimposed on a map like an old Disney movie. Then I saw a few feathers removed and then I saw skin underneath. I didn't say, "Is this a bird?" There was no classification like that going on.

A: Right. There are two things to say about that. First, a warning, and then an answer. It's a little dangerous to place a lot of credibility in your subjective experience of what happens when you answer questions or understand sentences. If we're conscious of anything, that's just the tip of the iceberg. In fact, we can't even say if it's a real piece of the iceberg or some completely misleading side effect caused by the iceberg. So that's the warning.

Having said that, I think there's a very serious question about whether or not the knowledge structures underlying language are in fact the same knowledge structures underlying visual information processing. If they aren't, then we should worry about which aspects of common sense reasoning would be better served by which structures.

And as far as I can tell, there's precious little interaction between high-level vision researchers and knowledge-based language researchers. This is too bad. Surely we both have needs related to spatial reasoning, although those concerns are probably much more central to vision processing than language processing.

There's been a certain amount of philosophical posturing around this question. Pylyshyn and Jackendoff come to mind. But it seems silly to jump to any conclusions given how little we really know about the whole business. I can't even say the jury is still out since the matter hasn't really come to trial.

The balloon was originally inflated.
The balloon broke (not the light bulb)
The light bulb was hot.
The light bulb was on.
* The heat caused the balloon the break.
* The balloon exploded.
* The explosion made a loud noise.
⊘ The baby was scared.
* The loud noise scared the baby.
* The baby cried because it was scared.
⊘ Mary is mad at John.
Mary communicated her anger to John.
⊘ Mary picked up the baby to comfort it.
⊘ John is not overly concerned
⊘ John will throw the balloon away.
* John was responsible for the balloon breaking.
* John was responsible for the baby crying.
* Mary is mad at John for making the baby cry.

* causal connections
⊘ goal states/emotional states

Figure 1. Inferences from the Balloon Story

THE BALLOON SCRIPT

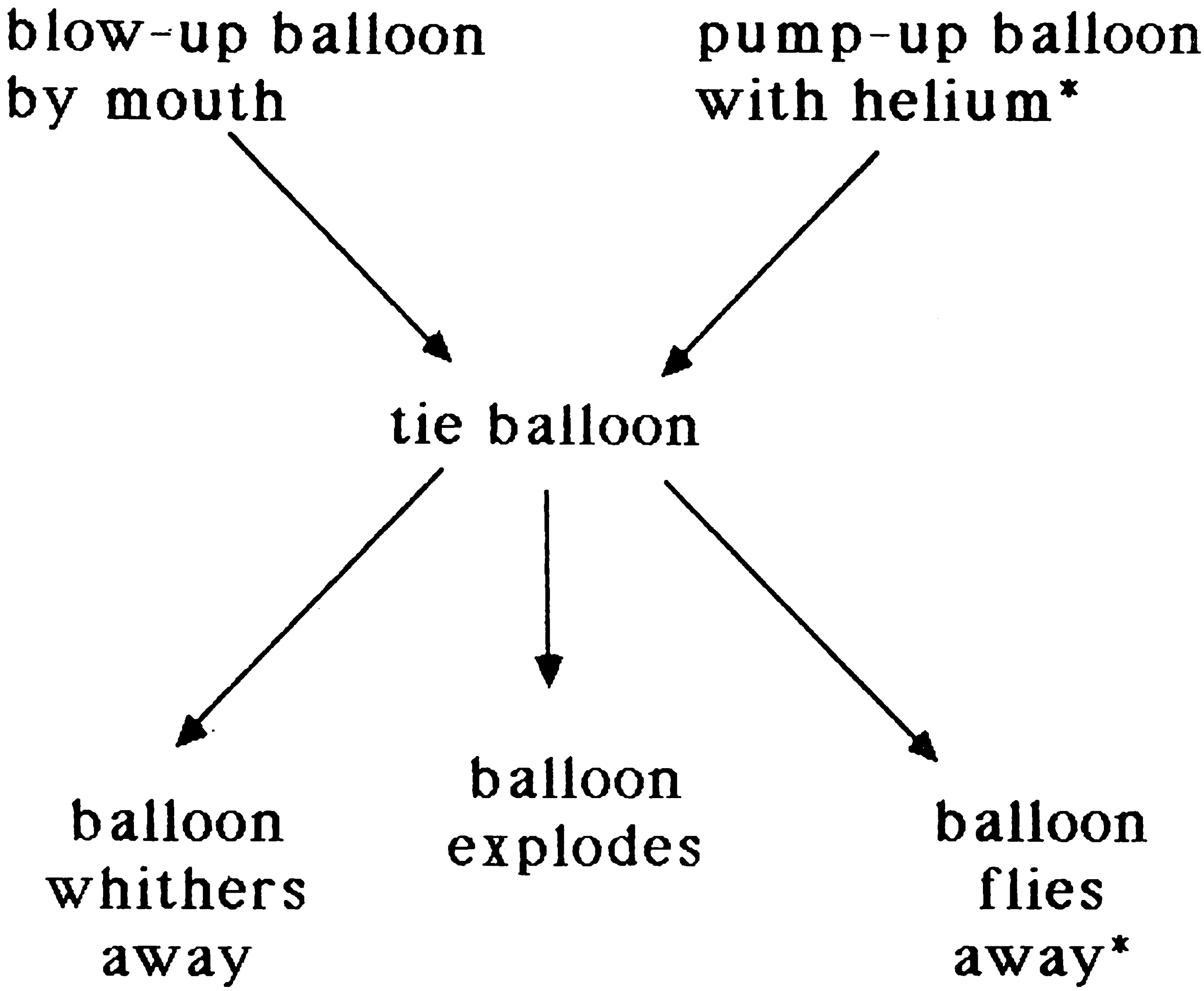


Figure 2. The Balloon Script

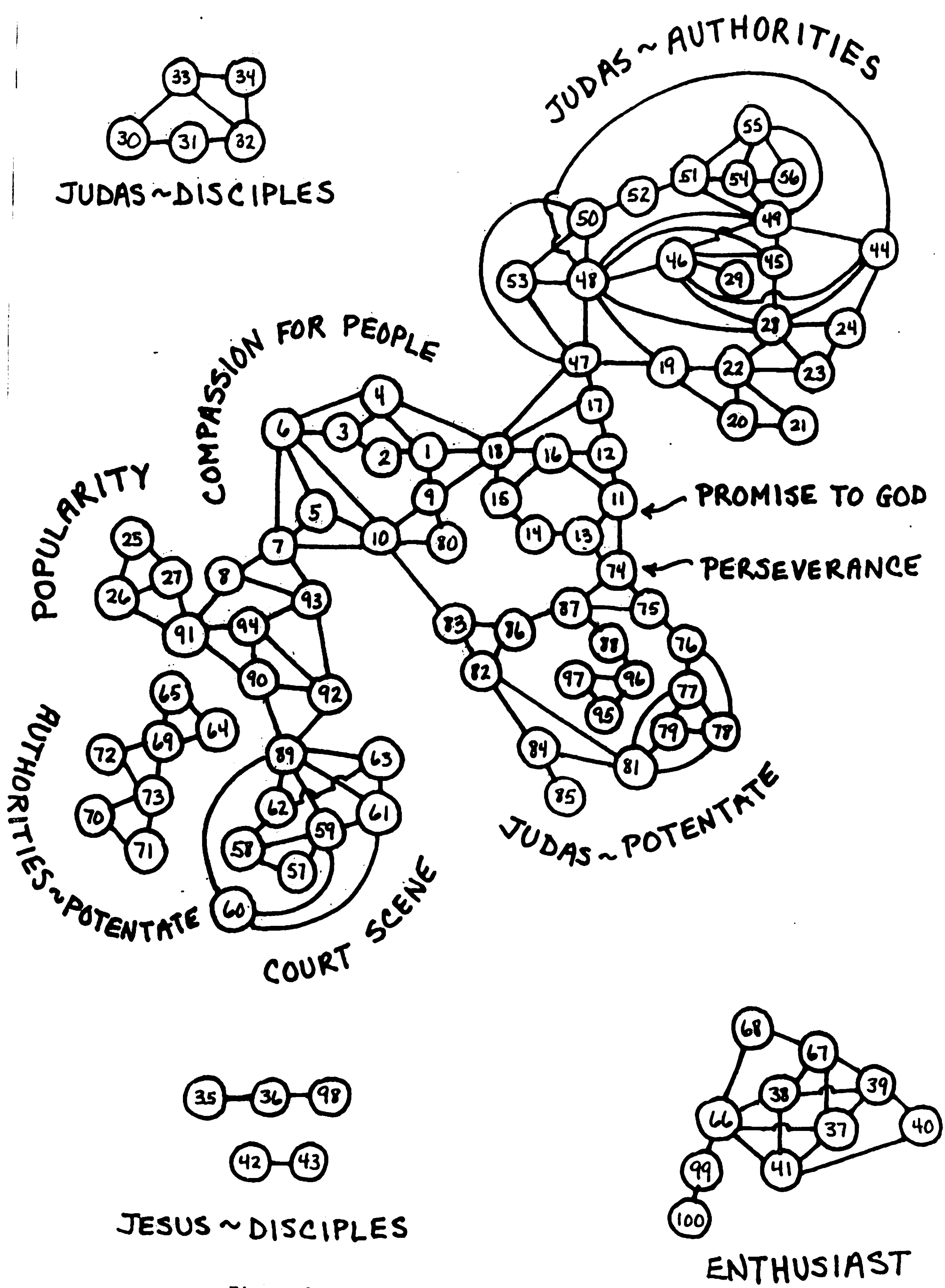
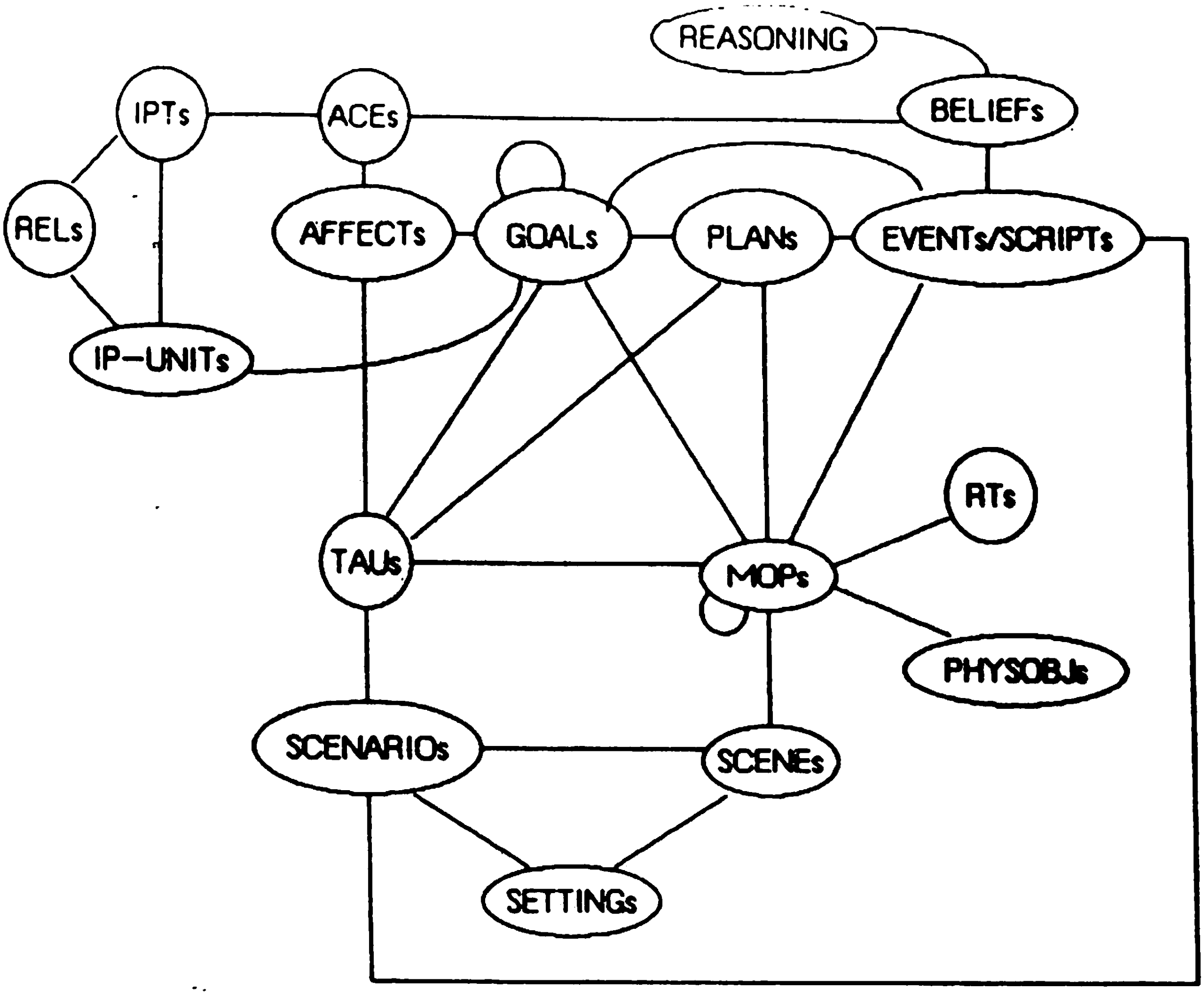


Figure 3. The New Testament in a Plot Unit Graph



IPT = Interpersonal Theme

IP-UNIT = Interpersonal Action

RT = Role Theme

REL = Relationship

MOP = Memory Organization Packet

TAU = Thematic Abstraction Unit

ACE = Affect as a Consequence of Empathy

KNOWLEDGE DEPENDENCY GRAPH

Figure 4. The Knowledge Dependency Graph for BORIS

INPUT:

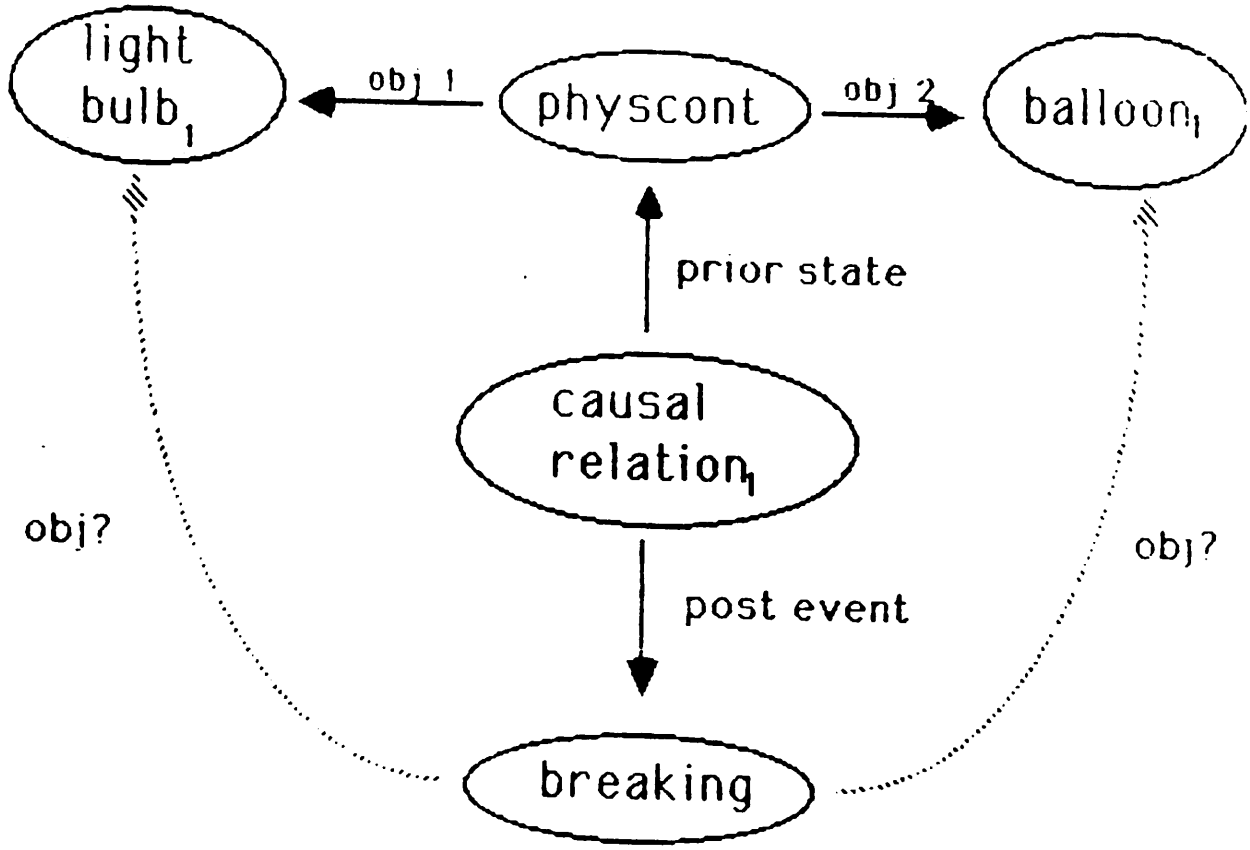


Figure 5. When the balloon touched the light bulb, it broke

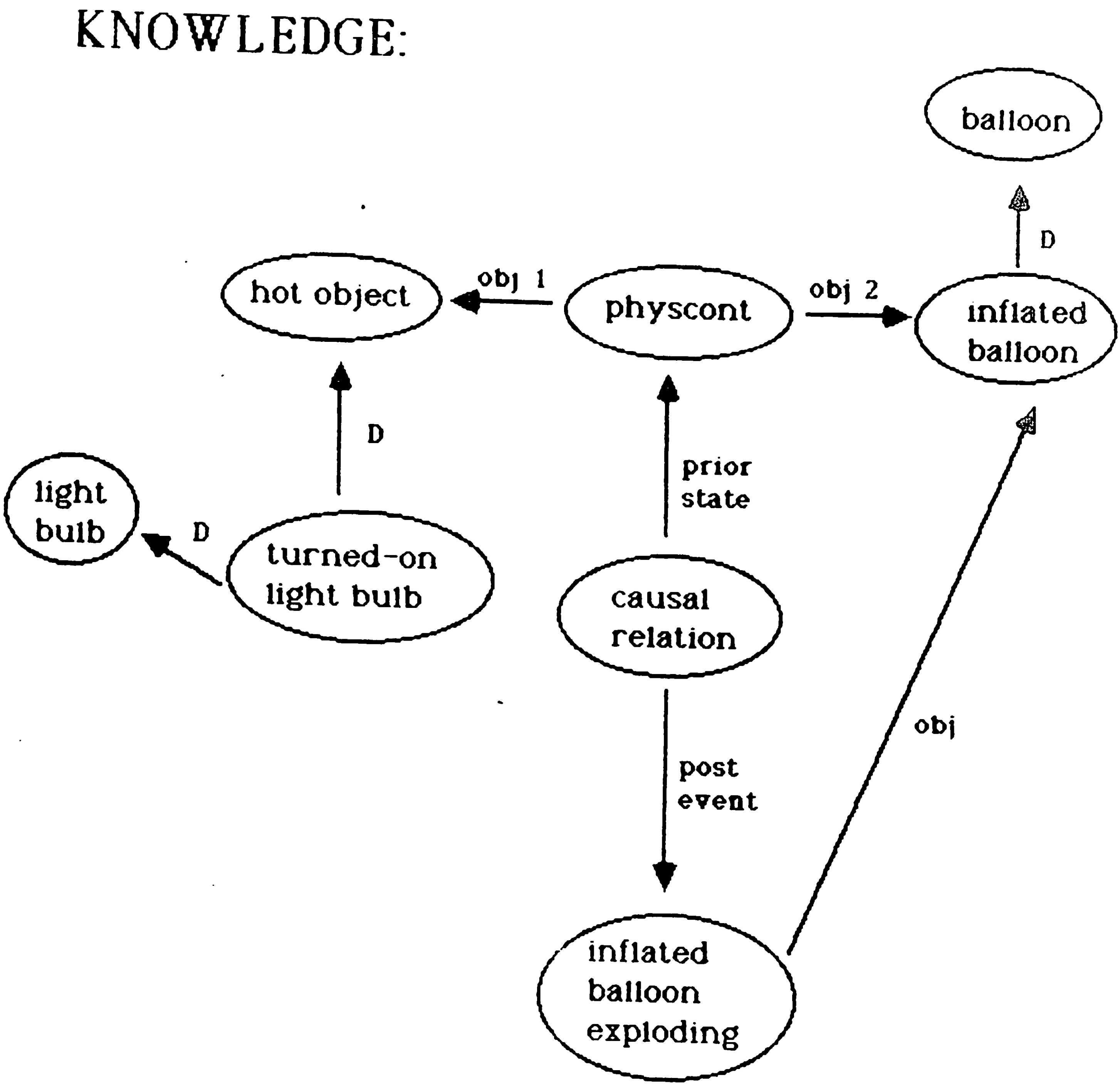


Figure 6. Inheritances for Exploding Balloons

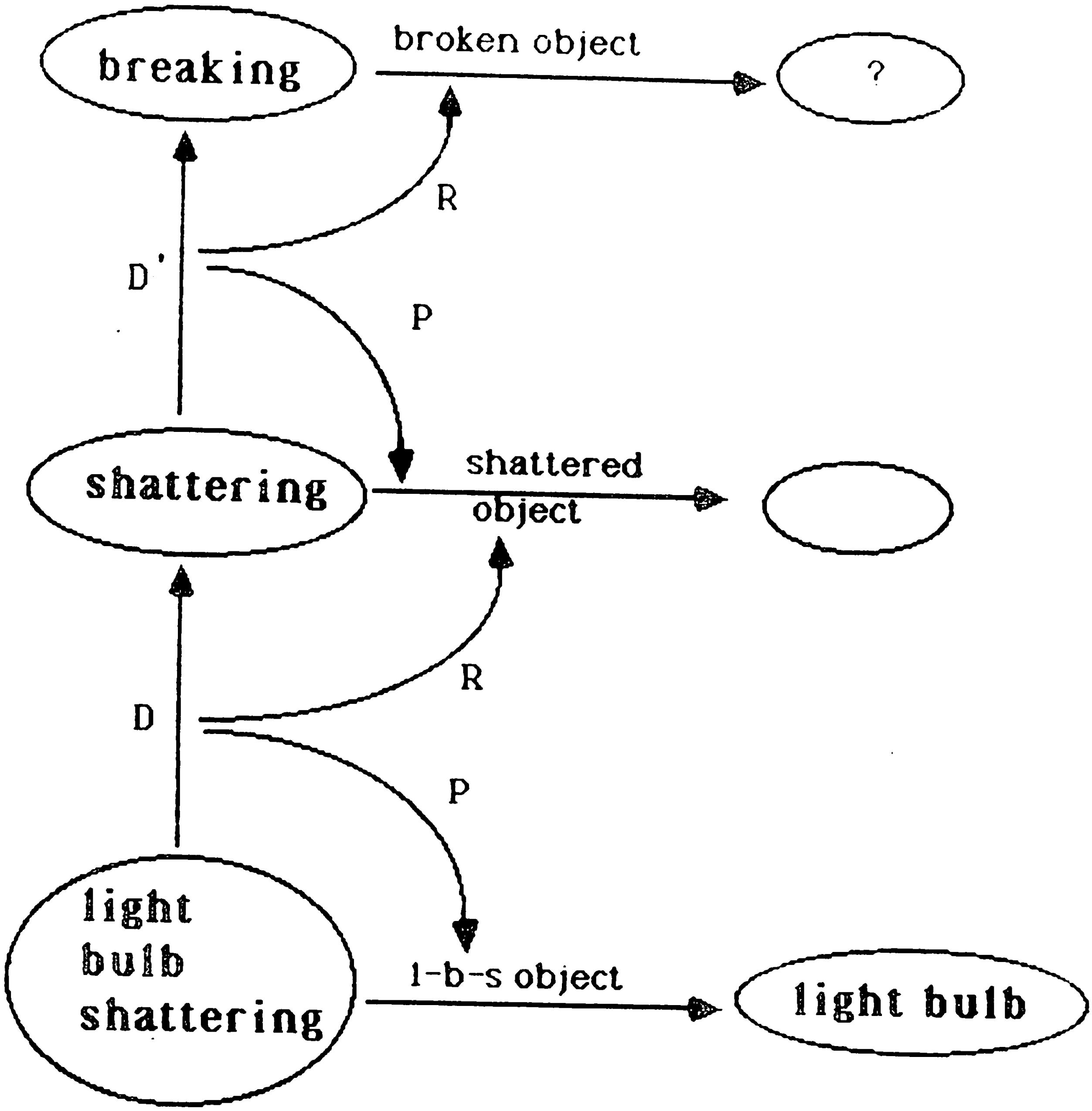


Figure 7. Inheritances for Shattering Light Bulbs

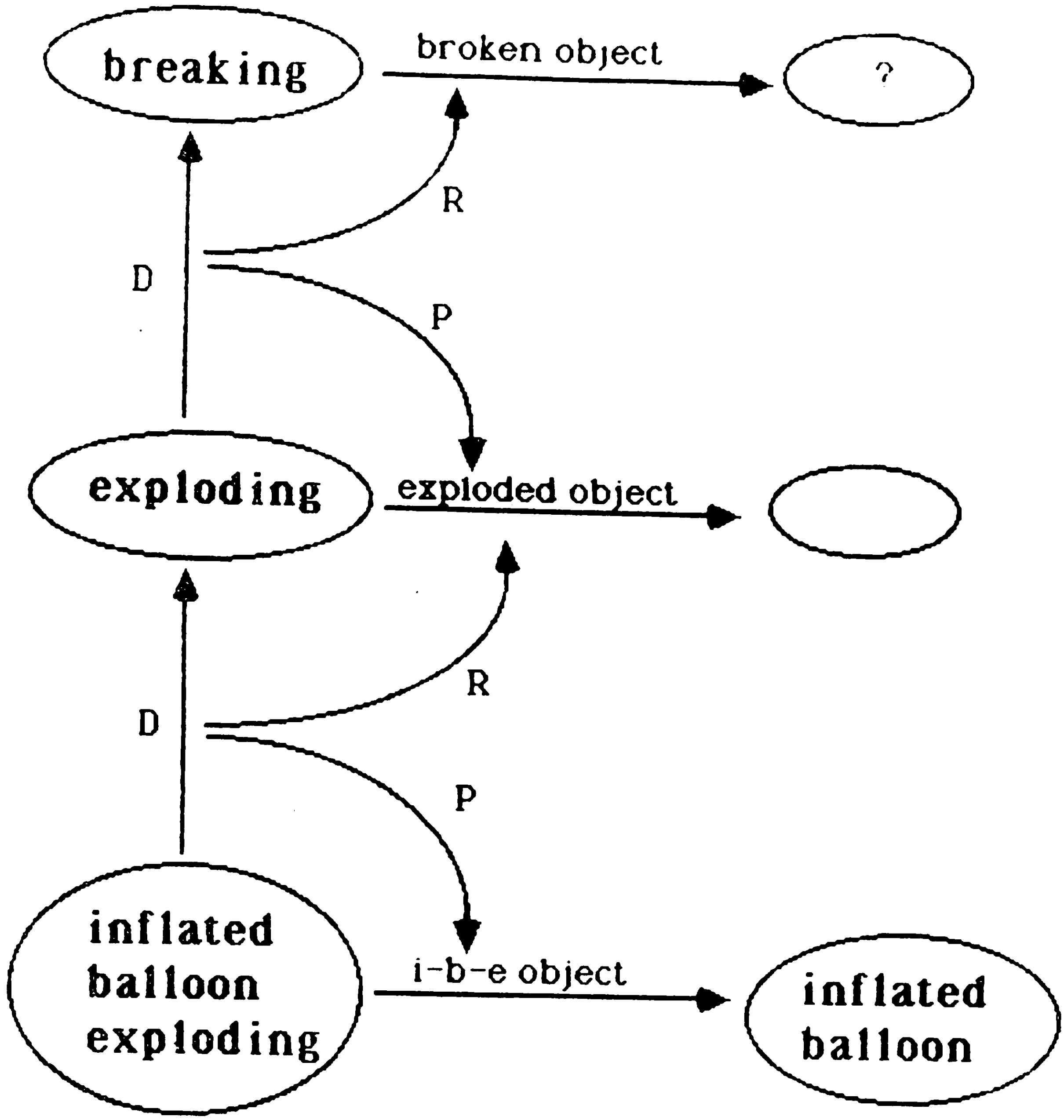


Figure 8. Knowledge about Balloons

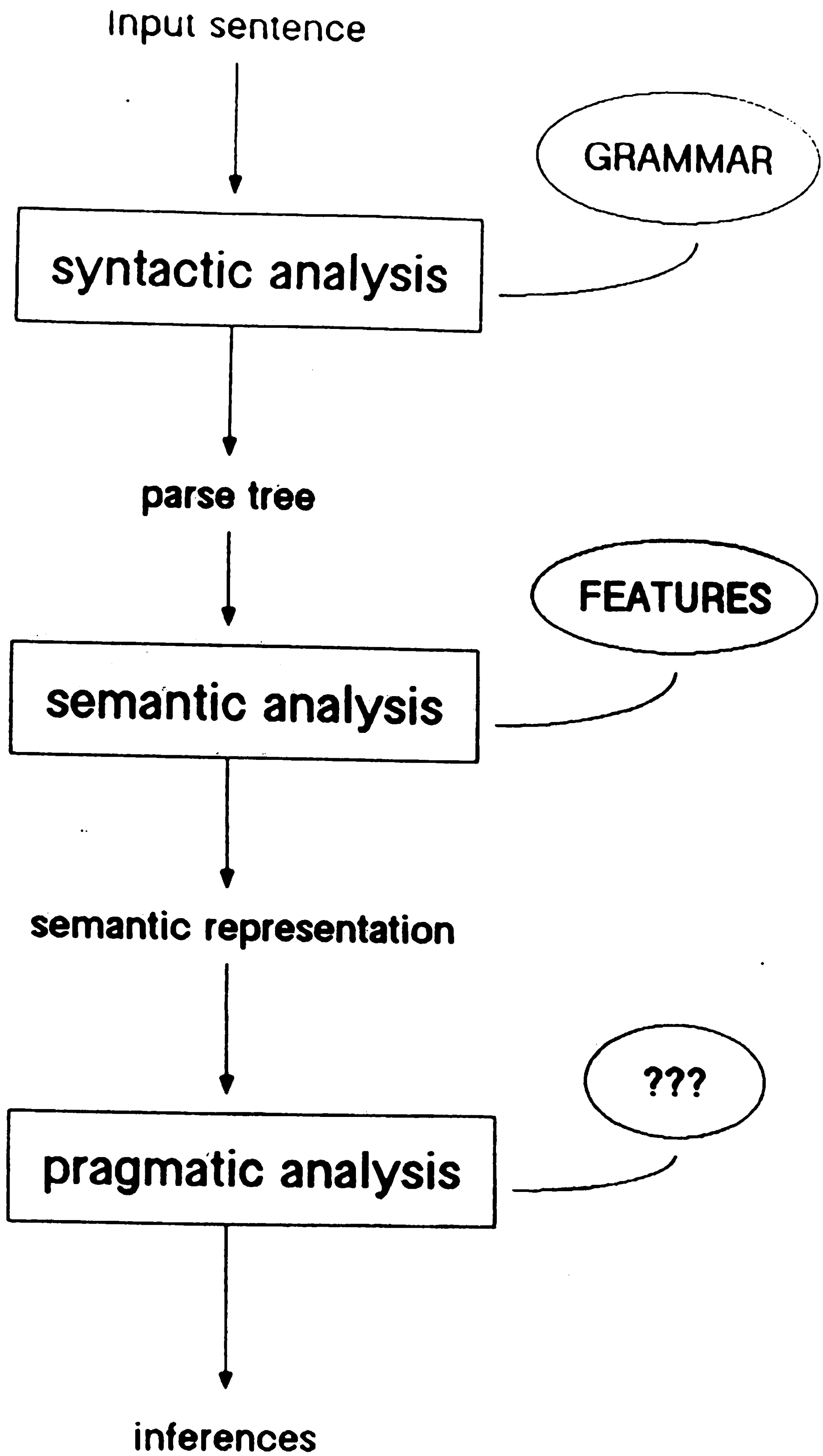


Figure 9. Serial Flow of Control

Mary was in the hospital.

John took her flowers.

(John took flowers to Mary)

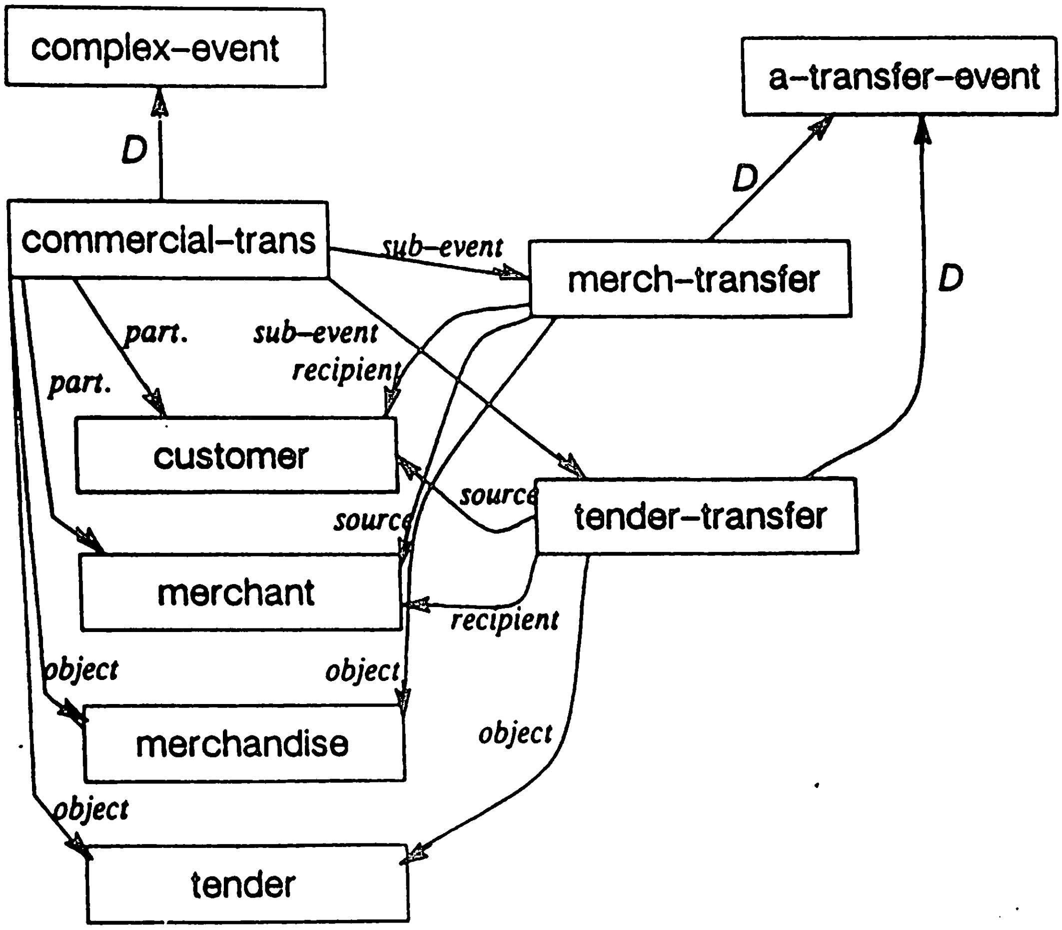
Mary was walking through Central Park.

A stranger took her money.

(A stranger took money from Mary)

Figure 10. Context Effects for Sentence Analysis

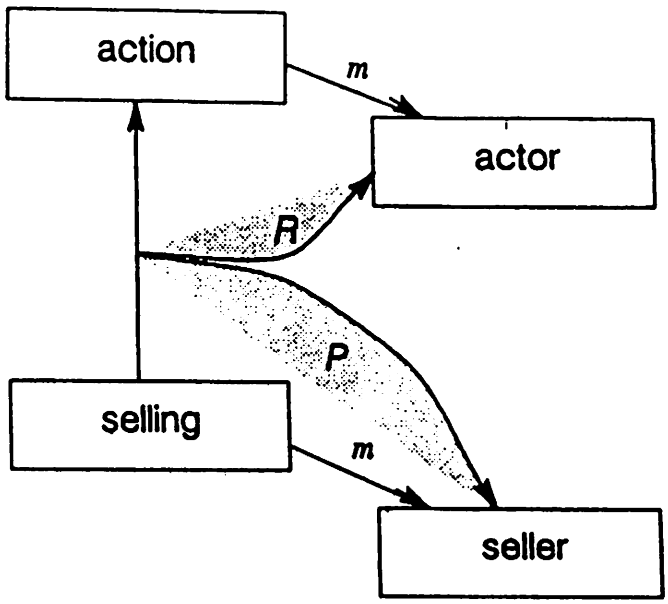
Structured Inheritance



1986:PSJ

Figure 11. Representing the verb "to sell"

Structured Inheritance



OR

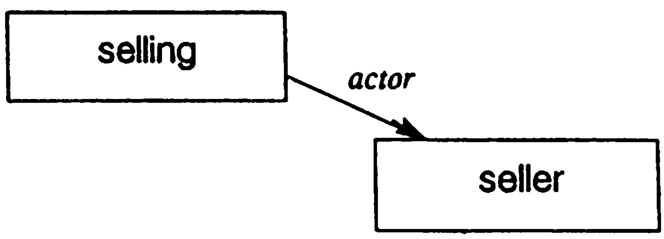


Figure 12. Implicit Role-Play Links

Putting it Together

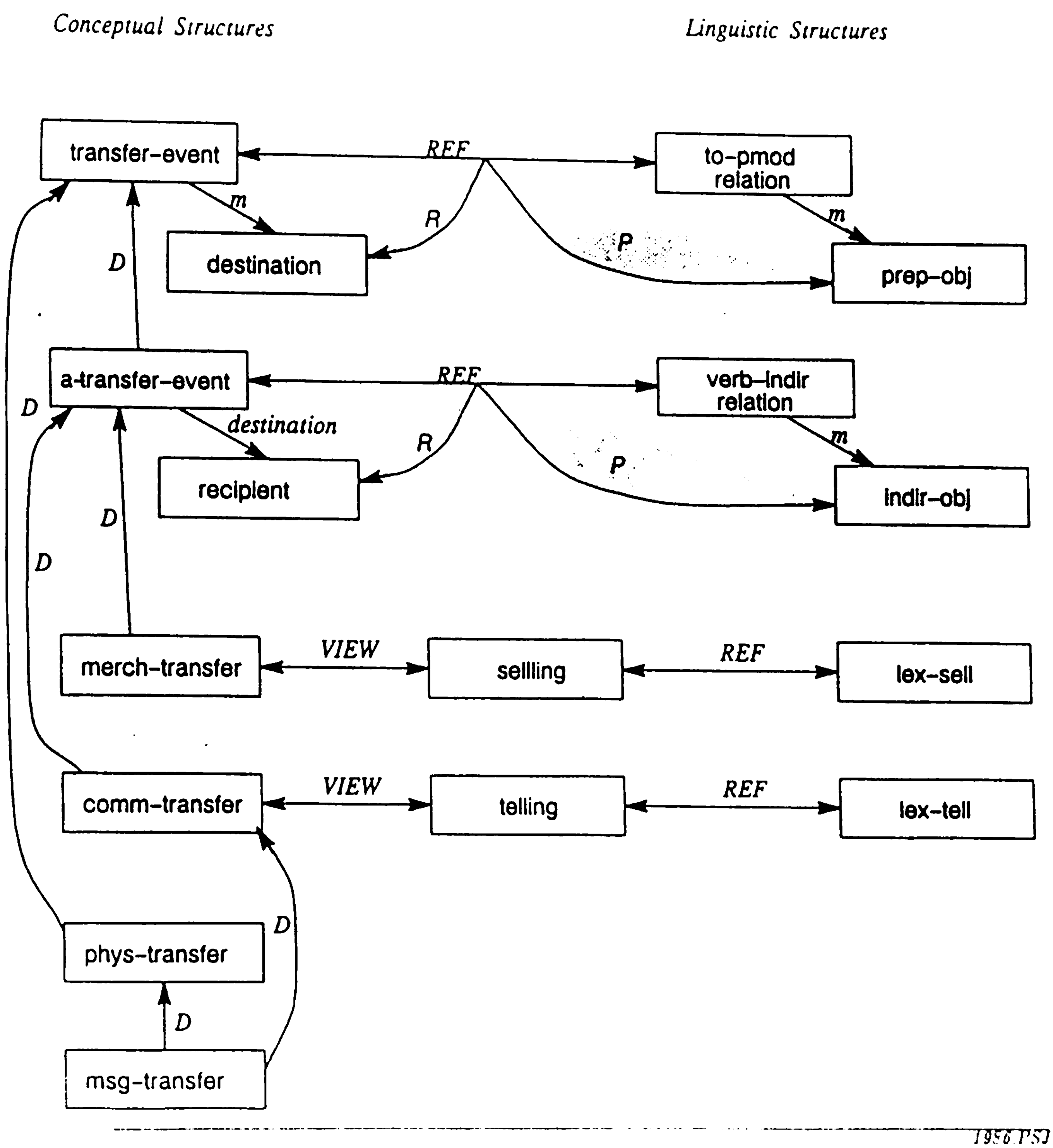


Figure 13. Integrating Syntax and Semantics

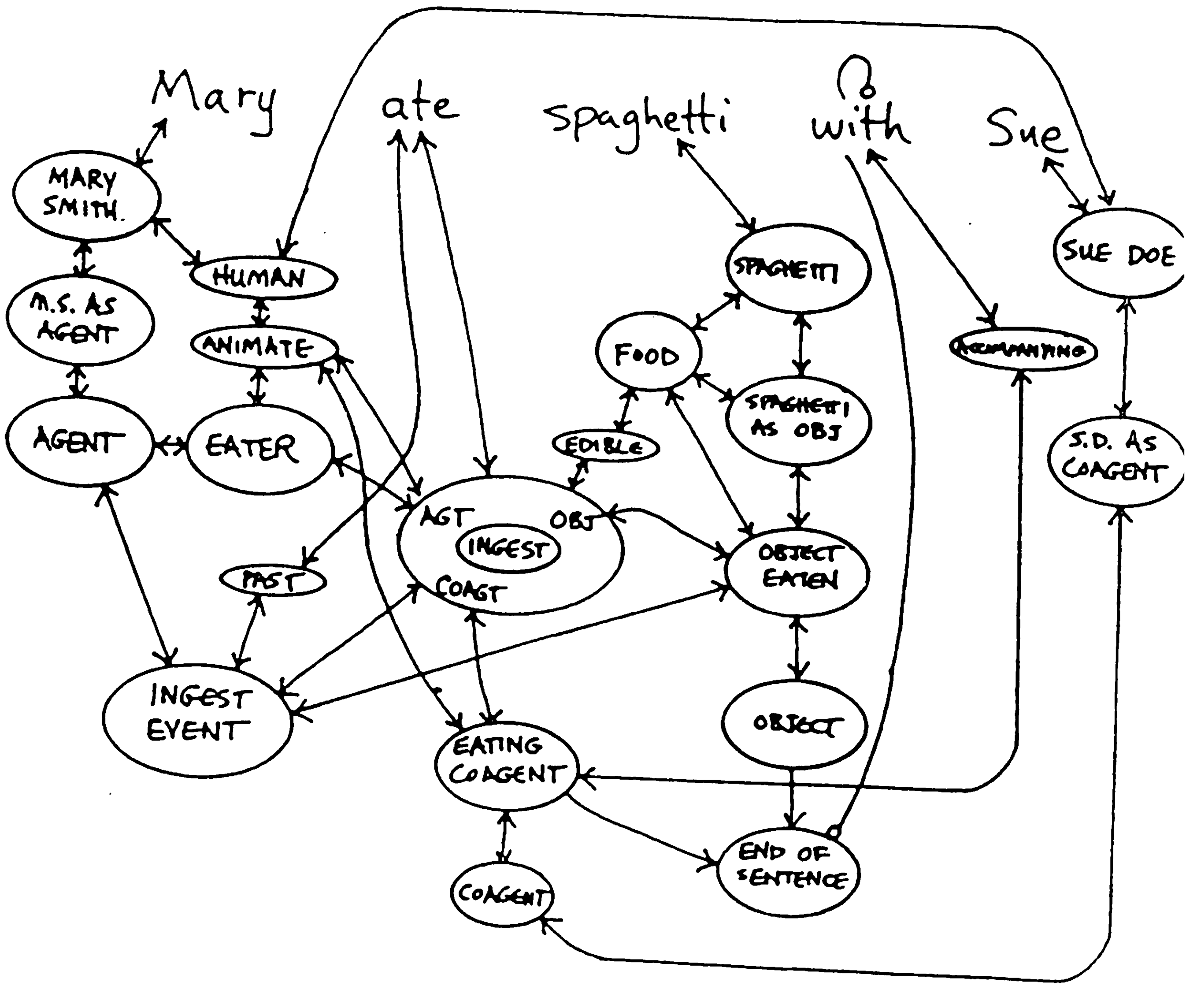


Figure 14. Eating Spaghetti with Massive Parallelism

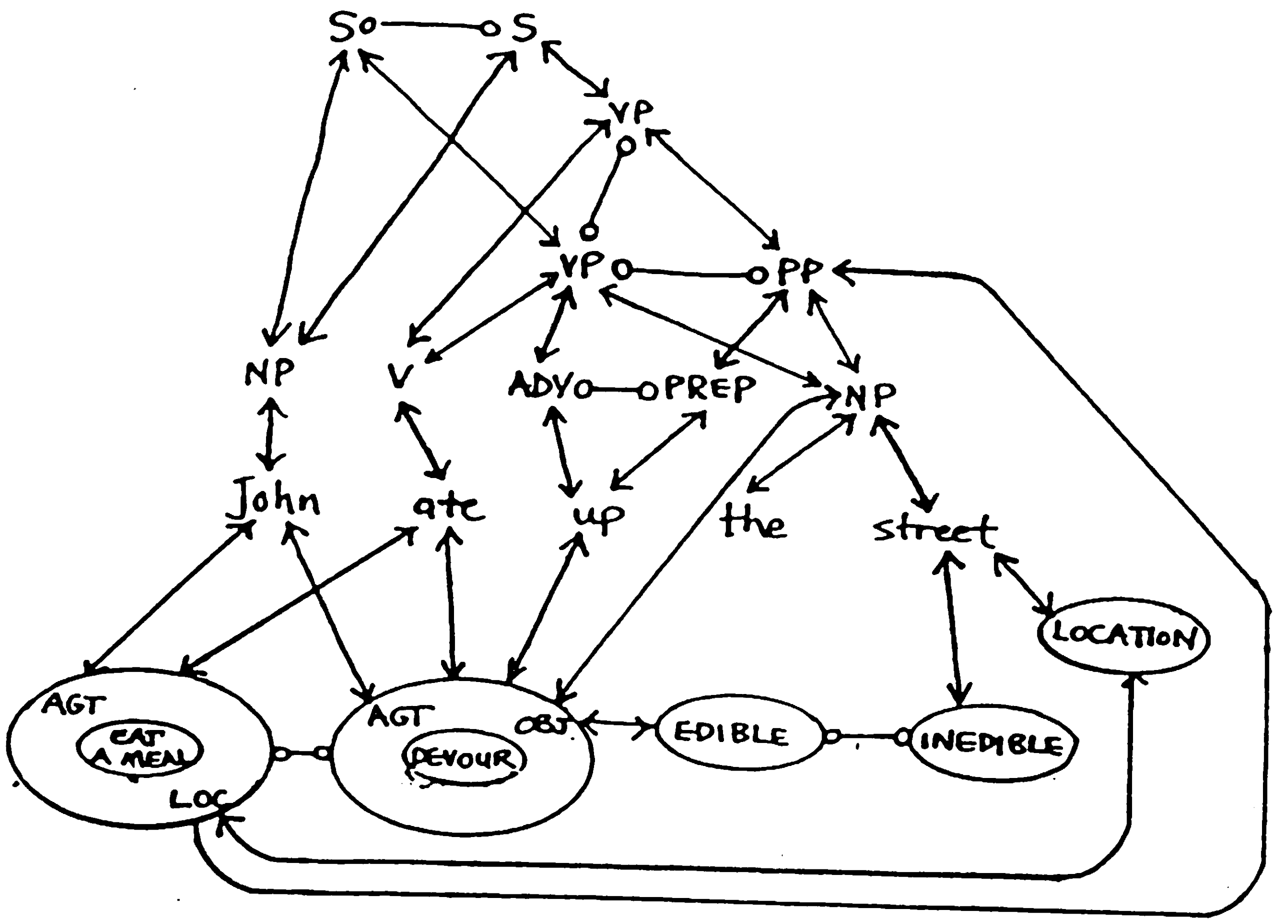
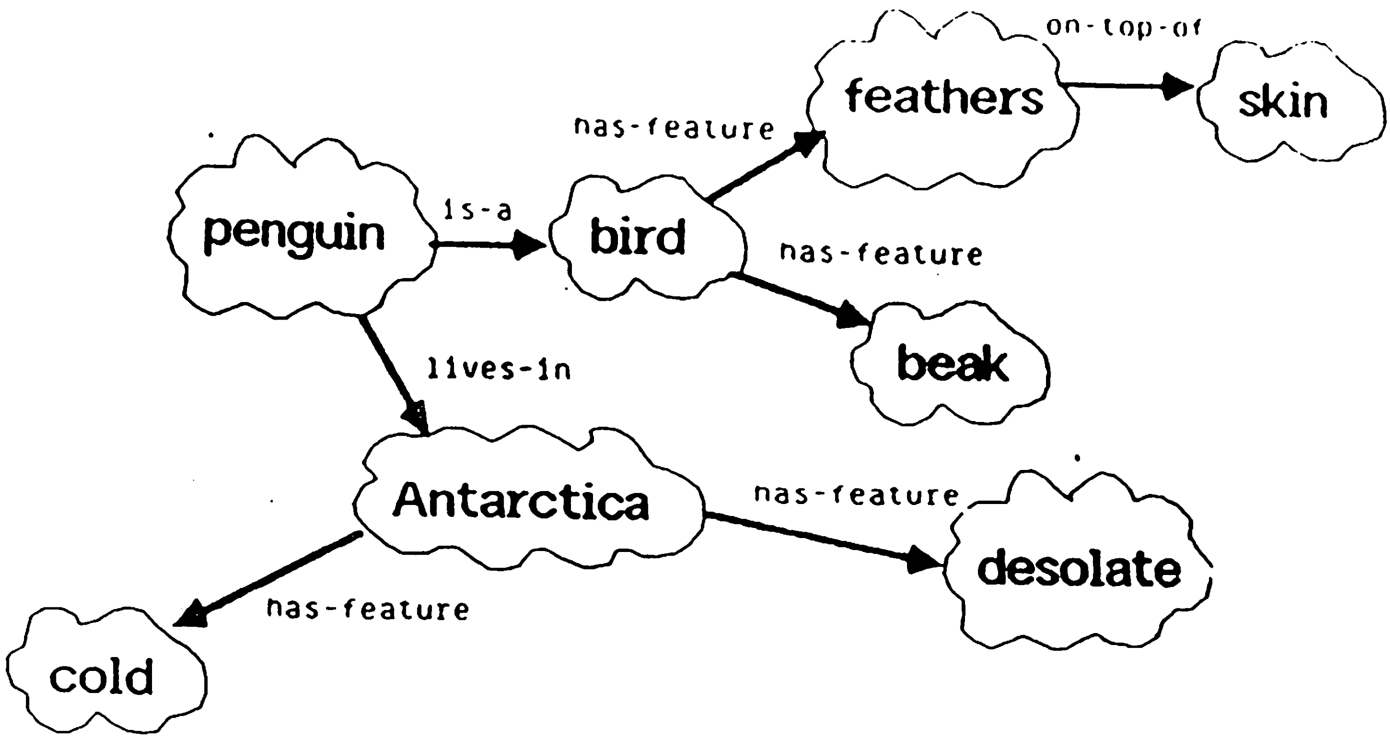


Figure 15. Adding Syntactic Constraints

Semantic Memory vs. Episodic Memory

Does a penguin have skin?



Does a chicken have skin?

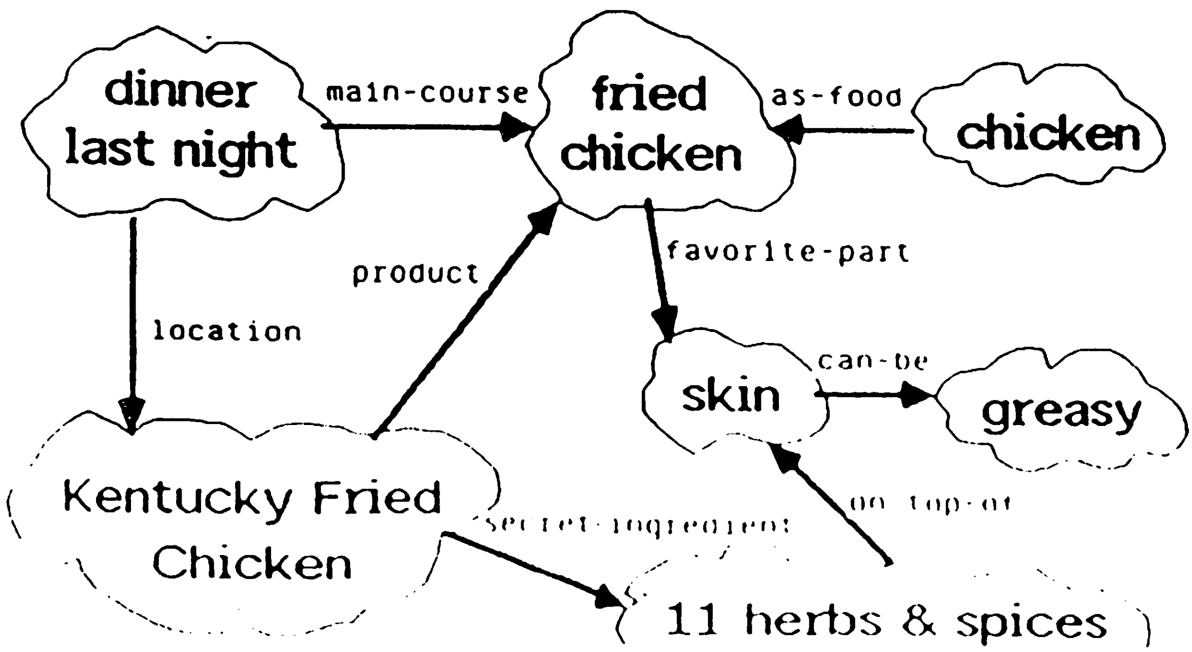
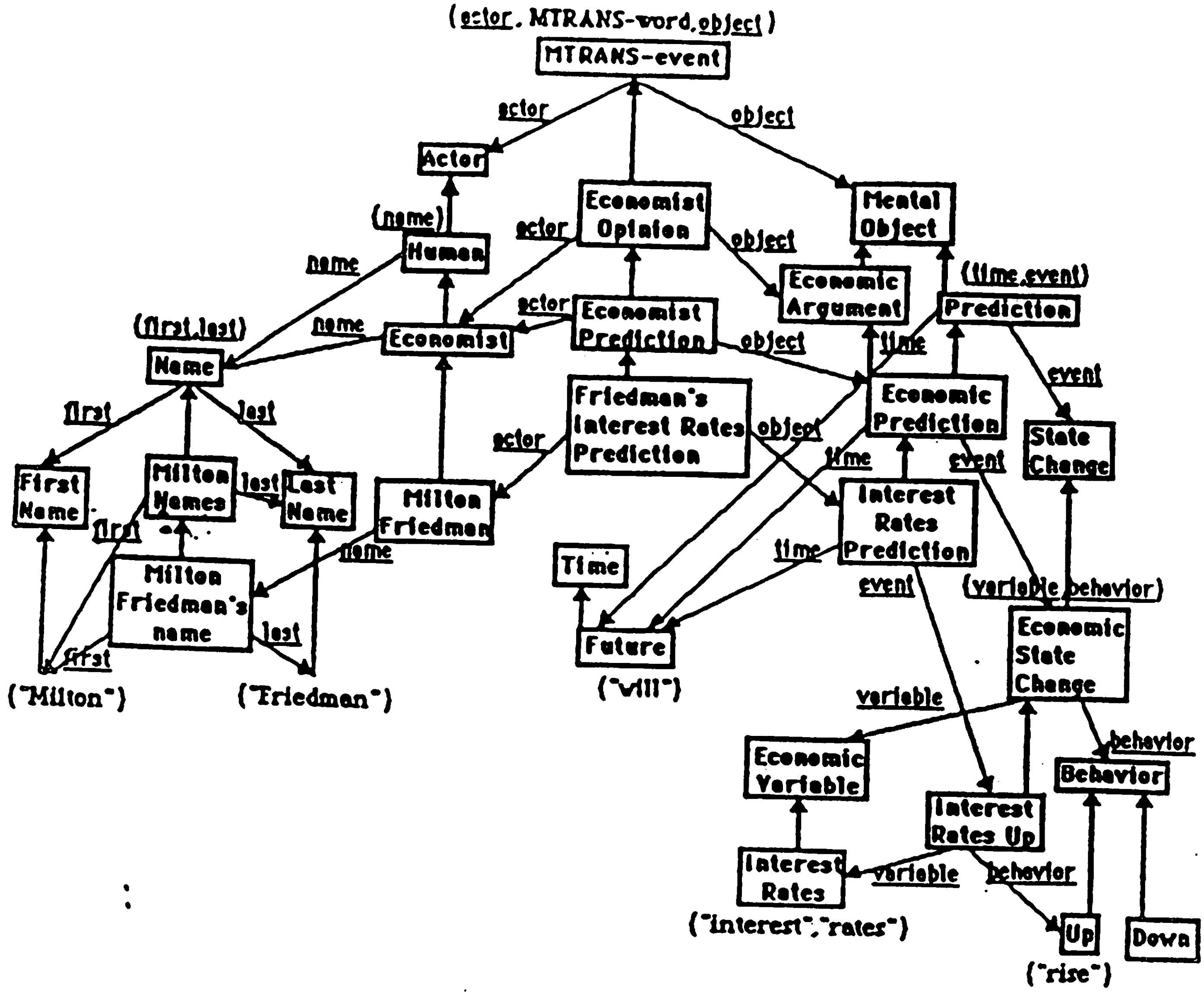


Figure 16. Semantic Memory vs. Episodic Memory



"Interest rates will rise as an inevitable consequence of the monetary explosion."

–Milton Friedman
(The New York Times, Aug. 4, 1984)

Figure 17. Understanding Milton Friedman