Title
Inference processing and error recovery in sentence understanding

Author
Eiselt, Kurt Paul

Publication Date
1989

Peer reviewed

UNIVERSITY OF CALIFORNIA

Irvine

Inference Processing and Error Recovery in Sentence Understanding

Technical Report 89-24

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Information and Computer Science

by

Kurt Paul Eiselt

Committee in charge:

Professor Richard H. Granger, Jr., Chair

Professor Edward H. Matthei

Professor Michael J. Pazzani

1989

The dissertation of Kurt Paul Eiselt is approved,
and is acceptable in quality and form for
publication on microfilm:

____

____

_______________________ Committee Chair

University of California, Irvine

1989

This dissertation is dedicated to my father
Robert Paul Eiselt

Contents

List of Tables

List of Figures

Acknowledgments

Many people have helped me with this work in many ways. My wife and best friend, Kathleen, has inspired, encouraged, and supported me in this endeavor since the day we met. My mother, Carol, and my grandmother, Ruth, have backed me in everything I have attempted for as long as I can remember, and my extended stay in graduate school was no exception. My brother, Kevin, who is the world's nicest guy, provided financial support on demand and took me on great camping trips, giving me much needed breaks from the grad school grind. My mother-in-law, June, was always willing to help in any way possible, and my father-in-law, Alan, never asked, "When are you going to get a real job?" Thank you all for your love and support.

My friends outside of grad school also helped me along the way, especially Mike Wintermute and Joyce McCabe, Rich and Shannon Hakesley, and Josh Bleier. I have neglected you terribly while I was writing this dissertation, and I apologize.

Early in my graduate career I had the good fortune to meet four grad students who took me under their collective wing and taught me the fundamentals of grad school survival: John Conery, Steve Fickas, Gene Fisher, and Bruce Porter. Also deserving of special mention are Doug Fisher, Rogers Hall, and Ken Steinhoff, who roomed with me at different times despite my lackadaisical attitude toward housekeeping, and Dale McNulty, who has tolerated me as an officemate for many years.

The administrative staff of the ICS Department at UCI, besides being genuinely nice and a whole lot of fun, helped me to overcome the bureaucratic obstacles. Notable in this regard are Susan Hyatt, Sue Rose, Phyllis Siegel, Lynn Harris, Pat Harris, and Mary Day. Working in the artificial intelligence project gave me the added advantage of having secretarial support and entertaining conversation provided by Caroline Ehrlich, Laura Yoklavich, Audrey Weinland, and Karen Toukdarian. The department's computing support staff, especially Jonathan Markson and Hieu Tran, helped keep my tired old workstation alive almost until the very end.

Rick Granger stirred my interest in natural language understanding when I first entered grad school, and has served admirably as my thesis advisor. Any success I have had in my graduate career is due in large part to his guidance. Ed Matthei of the Department of Social Sciences also served on my thesis committee and provided psycholinguistic expertise, suggestions on both the content and style of this dissertation, and good ideas for future research. Mike Pazzani, who came to my rescue by agreeing to serve on the thesis committee when another member had to bow out, also offered helpful advice for improving this dissertation.

Jeff Schlimmer of Carnegie Mellon University served above and beyond the call of duty or friendship. While trying to get his own postdoctoral career started, he carefully read through two drafts of this dissertation from cover to cover, offering suggestions on everything from the theory and its implementation to spelling and punctuation. Kadri Krause provided the induction proof in Appendix B. David Lewis of the University of Massachusetts at Amherst read several chapters and furnished useful feedback.

There is one person, however, whose contribution to this dissertation far exceeds all others. Jennifer Holbrook first postulated the theory of conditional retention, which is the keystone of the work reported herein, and she is primarily responsible for the experimental efforts to substantiate the theory. She convinced me of the merits of an interdisciplinary approach to natural language understanding, and has taught me most of what I know about psycholinguistics, experimental design, and data analysis. She has offered countless ideas for my research, and listened to countless others of my own. She has celebrated with me when things were going well, and consoled me when they weren't. She has graciously given up many nights with her family and taken time from her own doctoral studies so that she could help me with my work. I am truly fortunate to have so talented a colleague and so loyal a friend, and I am forever indebted to her.

Most of the work reflected in this dissertation was generously funded by National Science Foundation grant IST-85-12419. The remainder of the work was funded at various times by NSF grant IST-81-20685, and by Naval Ocean Systems Center contracts N00123-81-C-1078 and N66001-83-C-0255. I have also received financial support from the University of California, Irvine through a Regents' Dissertation Fellowship, and from Trilog Inc. through the Joseph J. Fischer Memorial Endowed Fellowship in Information and Computer Science.

Curriculum Vitae
Kurt Paul Eiselt

November 15, 1955	Born Fairbanks, Alaska
1976	B.S. in Information and Computer Science, University of California, Irvine
1976–1977	Programmer, NCR Corporation, Rancho Bernardo, California
1977–1980	Programmer and Customer Support Manager, Interactive Computer Systems, Inc., Costa Mesa, California
1980–1985	Teaching Assistant and Associate, Department of Information and Computer Science, University of California, Irvine
1982–1988	Research Assistant, Department of Information and Computer Science, University of California, Irvine
1983	M.S. in Information and Computer Science, University of California, Irvine
1988-1989	Computer Operator, Department of Information and Computer Science, University of California, Irvine
1989	Ph.D. in Information and Computer Science, University of California, Irvine Dissertation: *Inference Processing and Error Recovery in Sentence Understanding*

Publications

"STRATEGIST: A program that models strategy-driven and content-driven inference behavior," by R.H. Granger, K.P. Eiselt, and J.K. Holbrook. *Proceedings of the National Conference on Artificial Intelligence*, pp. 139-147, 1983.

"Interaction effects between word-level and text-level inferences: On-line processing of ambiguous words in context," by R.H. Granger, J.K. Holbrook, and K.P. Eiselt. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, pp. 172-178, 1984.

"The parallel organization of lexical, syntactic, and pragmatic inference processes," by R.H. Granger, K.P. Eiselt, and J.K. Holbrook. *Proceedings of the First Annual Workshop on Theoretical Issues in Conceptual Information Processing*, pp. 97-106, 1984.

"A parallel-process model of on-line inference processing." *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 863-869, 1985.

"Parsing with parallelism: A spreading-activation model of inference processing during text understanding," by R.H. Granger, K.P. Eiselt, and J.K. Holbrook. In J.L. Kolodner and C.K. Riesbeck (Eds.), *Experience, Memory, and Reasoning*, pp. 227-246. Hillsdale, NJ: Lawrence Erlbaum, 1986.

"A time-dependent distributed processing model of strategy-driven inference behavior," by K.P. Eiselt and R.H. Granger, Jr. *Program of the Ninth Annual Conference of the Cognitive Science Society*, pp. 704-714, 1987.

"Recovering from erroneous inferences." *Proceedings AAAI-87 Sixth National Conference on Artificial Intelligence*, pp. 540-544, 1987.

"(Almost) never letting go: Inference retention during text understanding," by J.K. Holbrook, K.P. Eiselt, R.H. Granger, Jr., and E.H. Matthei. In S.L. Small, G.W. Cottrell, and M.K. Tanenhaus (Eds.), *Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology, and Artificial Intelligence*, pp. 383-409. San Mateo, CA: Morgan Kaufmann, 1988.

Abstract of the Dissertation

Inference Processing and Error Recovery
in Sentence Understanding

by

Kurt Paul Eiselt

Doctor of Philosophy in Information and Computer Science

University of California, Irvine, 1989

Professor Richard H. Granger, Jr., Chair

Solving the mysteries of human language understanding inevitably requires an answer to the question of how the language understander resolves ambiguity, for human language is certainly ambiguous. But ambiguity leads to choices between possible explanations, and choice opens the door for mistakes. Unless we are willing to believe that the human language understander always makes the correct choice, any explanation of ambiguity resolution must be considered incomplete if it does not also account for recovery from an incorrect decision.

This dissertation describes a new approach to lexical ambiguity resolution during sentence understanding which is implemented in a program called ATLAST. Many computational models of natural language understanding have dealt with lexical ambiguity resolution, but ATLAST is one of the few models to address the associated problem of error recovery. ATLAST's ability to recover from an incorrect lexical inference decision stems from its ability to retain unchosen word meanings for a period of time after it selects the apparently context-appropriate meaning of an ambiguous word. The short-term retention of possible lexical inferences permits ATLAST

to recover from incorrect decisions without backtracking and reprocessing text, and without keeping a record of possible choices indefinitely.

The principle of retention provides a solution to the problem of error recovery which is compatible with current psycholinguistic theories of lexical disambiguation. Furthermore, the existence of some form of retention in lexical disambiguation is supported by the results of experiments with human subjects. This dissertation includes a discussion of these results and speculation on how the principle of retention might be extended to account for recovery from erroneous higher-level inference decisions.

Chapter 1

Scope of the Dissertation

1.1 The paradox of ambiguity

Ambiguity is essential to efficient communication with natural languages. Were we to communicate with our fellow natural language users so precisely that no ambiguity existed, we would find that communication was actually impaired by the myriad of tedious details which would necessarily be explicitly expressed in our text or speech. We economize by eliminating much of what could be said or written and rely on the listener or reader to supply the missing knowledge that is necessary to extract the intended meaning. We can never be sure, however, that the understander will supply the appropriate knowledge and arrive at the intended interpretation. Therein lies the paradox of ambiguity in language: we rely on ambiguity to make communication more efficient by decreasing the quantity of information being transmitted, but by doing so we also increase the potential for misunderstanding.

Fortunately, the human language understanding mechanism is very good at dealing with ambiguity. In fact, it is so good that we are seldom conscious that there is any ambiguity at all, although our daily communication is laden with multiple interpretations. This dissertation presents a theory of some of the processes involved in coping with ambiguity in language and a computational model called ATLAST which embodies that theory.

Ambiguity in language takes many forms, but the theory presented herein deals with only a subset of them. For example, the theory does not address problems of pronominal or syntactic ambiguity. The theory deals only with two types of ambiguity: *lexical ambiguity*, which arises from a word having more than one meaning, and *pragmatic ambiguity*, which occurs when a text implies more than one plausible sequence of events or state of affairs.

One example of lexical ambiguity comes from Swinney (1979):

Text 1: Rumor had it that, for years, the government building had been plagued with problems. The man was not surprised when he found several bugs in the corner of his room.

In this case, the language understander must decide whether the word "bugs" refers to insects or hidden microphones; usually the understander chooses the latter meaning because it seems more appropriate in the context established by the words "government building."

Pragmatic ambiguity, on the other hand, forces the language understander to make what ostensibly is a different kind of decision, as in this text from Schank and Riesbeck (1981):

Text 2: John went to a restaurant. He ordered chicken. He left a large tip.

Did John eat the chicken he ordered? Our knowledge of dining in restaurants tells us that he probably did so, though we can conceive of circumstances, such as a sudden illness or loss of appetite, which are not contradicted by the above story and which suggest the possibility that John did not eat the chicken.

ATLAST is able to make both kinds of decisions using a single mechanism for resolving ambiguity. In doing so, ATLAST accounts for the findings of psychological experiments investigating the nature of lexical access and disambiguation while accommodating recent theories about pragmatic inferences.

Many of the computational models of language understanding which have been proposed over the years are reviewed in the next chapter. These offerings represent only a small fraction of the natural language understanding systems which have been constructed in recent years, which suggests that a considerable amount of effort has been invested in solving the problem of ambiguity resolution. Many of these systems attack the problem of lexical ambiguity but ignore pragmatic ambiguity, while others deal with pragmatic ambiguity and ignore lexical ambiguity. ATLAST is one of the few systems which is capable of dealing with both kinds of ambiguity.

Another feature which sets ATLAST apart from other models is that it recognizes that a language understander makes mistakes. With very few exceptions, models of language understanding have assumed that although the input may be ambiguous, the understander always makes the correct decision the first time. This is of course false, as is demonstrated by the following text (inspired by Text 1):

Text 3: Officials at the U.S. Embassy in Moscow have called for a specialist to rid the new building of bugs. Secretaries there have reported seeing cockroaches in the employees' cafeteria.

Considering recent news reports about problems with the new U.S. Embassy building in Moscow (and the fact that the discussion of Text 1 above has biased the reader even if he or she is unaware of the problems), it is difficult to interpret the word "bugs" in Text 3 as meaning anything but hidden microphones on the first reading. By the time the reader sees the word "cockroaches" in the second sentence, however, the reader must revise the interpretation by supplanting the original choice with the insect meaning of "bugs."

ATLAST explains this ability to recover from an incorrect choice of word meaning with a process called *conditional retention* in which all meanings of an ambiguous word are retrieved, the meaning most appropriate to the preceding context is chosen, and the other meanings are temporarily retained. If later text contradicts the initially chosen meaning, the retained meanings are reconsidered in light of the updated context and a new meaning is selected. The conditional retention theory, which is supported by experimental evidence, is further extended in this dissertation to explain recovery from erroneous choices in resolving pragmatic ambiguity and to account for some individual differences in pragmatic ambiguity resolution.

1.2 Goals of the dissertation

As noted above, a great deal of effort has been devoted to solving the problem of ambiguity resolution in language understanding. One very simple and purely practical reason for this is that ambiguity is rampant in language, so understanding language requires the ability to resolve ambiguity. Another reason for addressing the problem of ambiguity resolution, one which motivates this dissertation, is that understanding ambiguity resolution may help us understand human cognitive processes.

Because the language understanding process lies "beneath our conscious awareness" (Carroll, 1986, p. 4), it does not readily lend itself to introspection. Thus, other investigative methods must be employed to gain insight into language understanding. Building a computational model of language understanding offers the opportunity to refine psychological theories by translating the theories into executable procedures.

Psychologists generally look to data from experiments with human subjects to determine the correctness of a theory, but a working computational model can serve to establish in advance the plausibility of that theory if the behavior of the model compares favorably to human behavior. The model-building process may also lead to predictions that are more specific and more easily tested through experiments with human subjects than those which may arise in the absence of a computational model. Furthermore, once built, a computational model provides a readily accessible framework for exploring changes to the theory before running new experiments with human subjects.

Although the techniques used in ATLAST may be useful in practical applications of language understanding systems, the goals that motivate the construction of this model arise from a desire to shed light on the human understanding process. These goals are:

- The model should demonstrate the plausibility of conditional retention and the associated mechanism for recovery from incorrect choice of word meaning as a theory of lexical ambiguity resolution in humans.
- The model should demonstrate the plausibility of a uniform theory of inference processing which arises from extending the theory of lexical ambiguity resolution to inference decision processes at the pragmatic level of understanding.
- The model should serve as the source of ideas, predictions, and constraints for experimental work with human subjects.

In deciding whether ATLAST satisfies these goals, one must consider the simplifying assumptions that have been made in constructing the model. These assumptions are described in the following section.

1.3 ATLAST: Assumptions and overview

A model of any process carries with it a set of assumptions which make life easier for the model builder. One frequent assumption is that the process being modeled can be isolated from its environment in such a way that the model still provides useful information. Without this assumption, the complexity of the process and its interaction with its environment may be so great as to prevent the building of a model. This is certainly the case with models of language understanding: modelers often assume that language understanding can be separated from other cognitive processes. Furthermore, they often assume that individual components of language understanding can be isolated from other components with little or no detrimental effect on the model. Several assumptions of this nature have been made in building ATLAST:

- ATLAST is a language understander that exists in isolation from other cognitive processes. For example, there is no phonological or morphological analysis; strings from the input stream map directly onto corresponding strings in ATLAST's lexicon.
- ATLAST is a model of automatic or unconscious processes. It is not intended as an explanation of what happens when the text is so confusing or the contradictions so severe that disambiguation or error recovery requires attentional problem solving.
- ATLAST contains processes for both syntactic and semantic analysis, but the emphasis is on the latter. ATLAST does not address issues of syntactic ambiguity resolution, recovery from an incorrect syntactic decision, or the degree of interaction between syntactic and semantic processing. On the other hand, ATLAST is designed so that these issues may be explored in the future without drastic architectural revisions.

The assumptions listed above help to describe what ATLAST is not, but there are other assumptions that influence what ATLAST is. For instance, ATLAST assumes that semantic knowledge is organized in a relational network like that described by Quillian (1968) and later elaborated by by Collins and Loftus (1975). This memory consists of nodes and links; the nodes represent objects, events, or states, and the links represent relationships between the nodes. The details of the relationships stored in this memory are generally unimportant to ATLAST except that relationships of causality, intentionality, and abstraction are identifiable.

The assumption about memory in turn influences what constitutes an interpretation of a text. ATLAST's interpretation of a text consists of two parts: a set of connections or paths in memory that ties together all open class words[1] from the input text, and a set of pointers to nodes in memory that imposes a temporal ordering on the events and states of the text and indicates which nodes fill the thematic roles for those events and states. For example, when given this input:

```
John went to the pawnshop.
John sold a lamp.
```

ATLAST will determine that these paths in memory best explain the relationships between the meanings of the words in the text:

```
Path from LAMP to SELL
  LAMP is an instance of OBJECT
  OBJECT is a role-filler of POSSESS-OBJECT
  POSSESS-OBJECT is a precondition of SELL
Path from PAWNSHOP to GO
  PAWNSHOP is an instance of BUSINESS
  BUSINESS can be viewed as BUYER
  BUYER is a role-filler of BE-AT-BUYER
  BE-AT-BUYER is an instance of BE-AT-PLACE
```

[1]Nouns, verbs, adjectives, and adverbs are examples of open classes of words. The name derives from the fact that the number of members of these classes is unbounded. New names for objects, actions, and properties can be invented by generating new words or by adding endings to existing words. Closed classes, on the other hand, contain a fixed number of members; additions to the class occur rarely if at all. Prepositions and conjunctions are examples of closed classes. (Winograd, 1983)

```
  BE-AT-PLACE has the precondition GO
Path from SELL to JOHNO
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
  POOR-STATE is an instance of HUMAN-ECON-STATE
  HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
  GENERIC-HUMAN has the instance JOHNO
Path from SELL to GO
  SELL has the precondition BE-AT-BUYER
  BE-AT-BUYER is an instance of BE-AT-PLACE
  BE-AT-PLACE has the precondition GO
```

It will also determine that the thematic roles of the events explicitly stated in the input should be filled as follows:

```
Event:  event0
Actor:  (JOHNO)
Action: (GO)
Object: nil
Direction: (PAWNSHOP)

Event:  event1
Actor:  (JOHNO)
Action: (SELL)
Object: (LAMP)
Direction: nil
```

These two components of ATLAST's interpretation are also shown in Figure 1.1.

Finding the most explanatory set of paths is the responsibility of the semantic analysis component, while assignment of the pointers is done by the syntactic analysis component. Although the emphasis in this work is on semantic processing, neither component takes a back seat to the other when ATLAST is interpreting a text; both components contribute to a complete understanding of the input. The two components of the interpretation are generated through the combined efforts of three processes running concurrently: the *capsulizer*, the *proposer*, and the *filter*. The relationships between these components are shown in Figure 1.2.

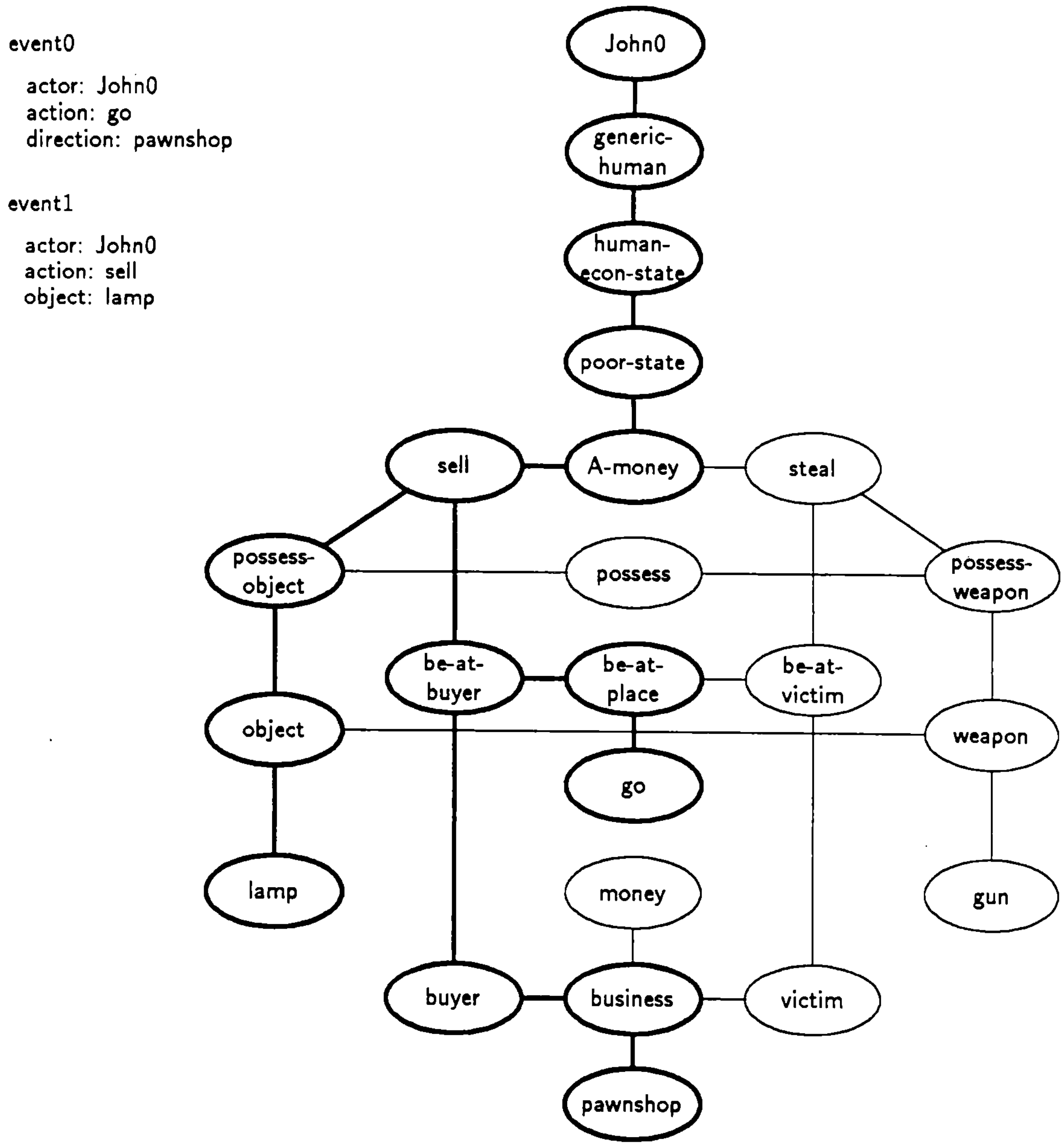

Figure 1.1: The two components of ATLAST's interpretation: event chain and active paths.

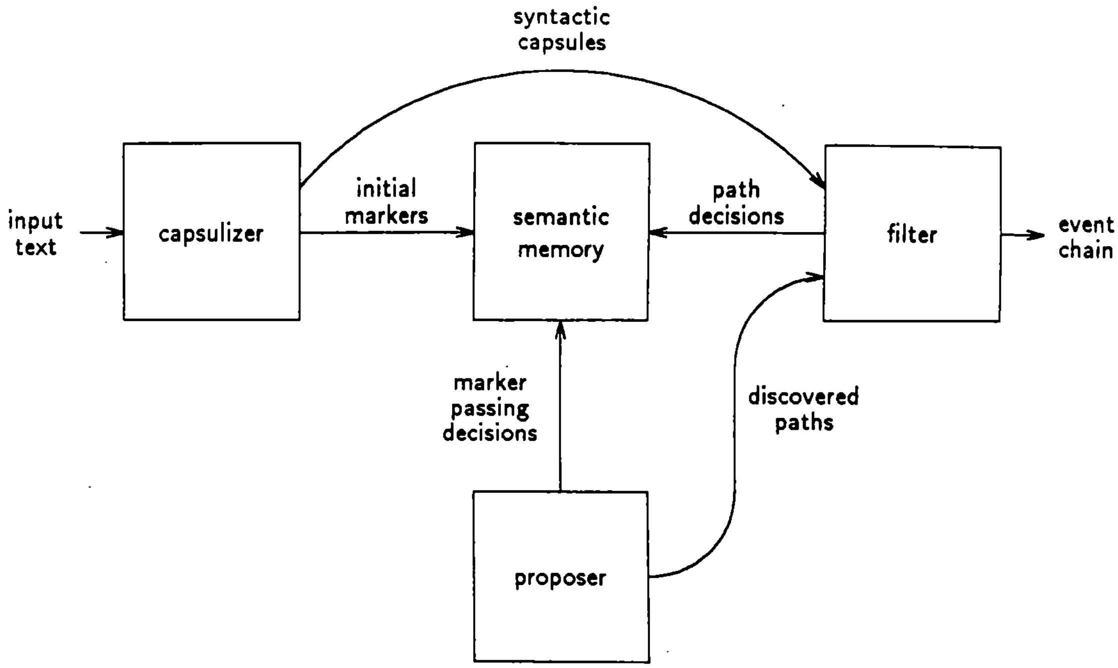

Figure 1.2: Relationships between ATLAST's processes.

The input is read one word at a time by the capsulizer. Upon reading a word, the capsulizer finds the word in ATLAST's lexicon and activates the meanings associated with the word. These activated word senses are the starting points for the marker-passing search carried out by another process, the proposer, which is introduced below. In addition to initializing the search for connecting paths in memory, the capsulizer accumulates the syntactic category information it retrieves from the lexicon and makes decisions about syntactic relationships within the constituent phrases of the input text. These *intra-phrasal* syntactic decisions are combined with pointers to the word senses activated by the words in the phrase and passed as capsules of information to a third process, the filter, which is also introduced below. If a word activates more than one word-sense, the pointers to the multiple word-senses are all passed on to the filter, which will eventually select the most plausible word-sense.

The proposer is a breadth-first search mechanism that uses marker-passing to traverse the links between the nodes in memory and find connections between word-senses which have been activated by the capsulizer. The proposer maintains pointers to the most recently activated nodes in memory, and to the word-senses which are the origins of the spreading activation search. Each time the proposer is invoked, it traverses the links leading away from the recently activated nodes, activates the adjacent nodes at the end of those links, and updates its list of pointers. If the spread of activation from one point of origin intersects the spread of activation from another point of origin, then the proposer has found some plausible relationship, by way of links and nodes, between two (and possibly more) of the word-senses activated by the input text. The proposer then passes information about this newly-discovered pathway to the filter.

The proposer is implemented in ATLAST as a separate process, but from a theoretical perspective it might be more appropriately viewed as an emergent property of human memory organization. Because computer memory works differently from human memory, it was necessary to provide a separate process to make the spreading activation possible.

The filter performs three functions, the first of which is *inter-phrasal* syntax. As capsules are passed from the capsulizer to the filter, the filter makes decisions about the relationships between the phrases represented by the capsules. Inter-phrasal syntax rules enable the filter to assign fillers to the various thematic roles such as actor, action, and object. The filter's second function is the evaluation of explanatory paths found by the proposer. When two competing paths are proposed (e.g., different paths connecting the word-senses of two words from the input text), the filter attempts to select the path more appropriate to the existing context through the application of inference evaluation metrics. Finally, the filter is also responsible for correcting ATLAST's interpretation if it discovers that a previously chosen path is contradicted by later text.

1.4 Overview of the dissertation

Marr (1982) has proposed three levels at which a machine performing an information processing task must be understood:

- **Computational theory:** The goal of the computation, the reason for the computation, and the logic of the strategy by which it can be performed.
- **Representation and algorithm:** The implementation of the computational theory—the representation of the input and output, and the algorithm for transforming one into the other.
- **Hardware implementation:** The physical realization of the representation and algorithm.

This dissertation describes a theory of ambiguity resolution and error recovery at the level of Marr's computational theory, and a computer model called ATLAST which corresponds to Marr's representation and algorithm level. There is no attempt here to speculate on how the representation and algorithm might be implemented in the hardware of the human brain.

ATLAST is only one of a number of representations and algorithms that could serve as functional realizations of the theory described herein, so some separation between the two levels has been maintained throughout this dissertation in order to avoid any confusion that might be caused when the program diverges from the theory. Maintaining this separation has led to an organization in which chapters that concentrate on theory alternate with chapters emphasizing the program.

Chapter 2 introduces four constraints which a model of language understanding should take into account. These constraints have been derived primarily from the results of psychological experiments on human subjects. Many previous models have accounted for a subset of these constraints, and a number of these models are reviewed in this chapter. ATLAST, however, is the only model which accounts for all four constraints. Chapter 3 contains a detailed description of ATLAST's architecture, which is directly influenced by these four constraints.

Chapter 4 explains the theory of conditional retention. Conditional retention extends widely-accepted theories of multiple access of word meanings to account for a human understander's ability to recover from an incorrect choice of word meaning without reprocessing the input. Chapter 5 illustrates how conditional retention enables error recovery through a detailed example in which ATLAST processes a misleading text.

Chapter 6 discusses how the theory of lexical disambiguation and error recovery described in Chapters 4 and 5 can be extended to account for pragmatic inference decisions. This chapter also shows how conditional retention offers an explanation of some individual differences in pragmatic inference behavior. Examples of ATLAST in operation using this extended theory are presented in Appendix A.

Up to this point the dissertation emphasizes how well ATLAST accounts for experimental data on human language understanding. Chapter 7 looks at ATLAST from another viewpoint, that of computational efficiency. There have been challenges to models using spreading activation or marker-passing suggesting that these search mechanisms are computationally inefficient. The analysis in Chapter 7 demonstrates that the use of computational resources can be held to an acceptable level through the application of appropriate constraints.

The dissertation concludes with Chapter 8. This chapter summarizes the key points of the model, discusses some open questions, and suggests directions for further research within the ATLAST framework.

Chapter 2

Constraints on Understanding

2.1 Perspectives

The artificial intelligence (AI) literature contains numerous examples of different research paradigms. One informative effort to find order in this confusing assortment of differing methodological approaches is offered by Hall and Kibler (1985). They divide AI research into two broad classes based on the intent of the researcher. Within one class, called the *artificial* perspective, researchers are concerned with studies of intelligent function regardless of the underlying mechanism. The other class of research, the *natural* perspective, is characterized by the investigation of human (or animal) cognitive phenomena. This dissertation follows the natural perspective, which is further subdivided along methodological lines into the *empirical* and *speculative* perspectives.[1] In the empirical perspective, researchers offer explicit evidence of the correspondence between the behavior of their models and the natural subjects being modeled. Researchers who follow the speculative approach make less rigorous efforts to provide empirical evidence supporting their models. Hall and Kibler are careful to note, however, that the empirical and speculative perspectives do not correspond to

[1]The artificial perspective is also further subdivided, but the results are less relevant to the current discussion.

"careful" and "reckless" approaches. Instead, these two perspectives represent differences in problem choice (narrowly-constrained, well-defined problems as opposed to representative examples of loosely-defined behavior), method of solution (experimentation and incremental modification of the model versus introspection and informal observation), and criteria for evaluation (experimentally obtained evidence of correspondence between the model and the human subject versus working programs that mimic human behavior on the examples).

Hall and Kibler also note that the different perspectives they describe are not mutually exclusive: ideas may be shared across perspectives, and individual researchers may freely shift between perspectives. The empirical and speculative approaches can be viewed as extremes on opposite poles of a continuum of perspectives. Thus, while ATLAST leans toward the speculative perspective, it borrows from the empirical perspective in that constraints on the model come not only from informal observations of human language behavior on both normal and aberrant texts but also from the results of psycholinguistic experiments.

The observations which have been made, both formal and informal, on the behavior of the human language understanding system are innumerable, the conclusions drawn from many of these observations conflict with other conclusions, and no computational model of language understanding can be expected to account for them all. It is not difficult, however, to construct a small subset of compatible, reasonably well-founded observations that provide constraints on the design of natural language understanding (NLU) models—phenomena that any computational model should be able to explain. This chapter presents one such set of constraints, describes a number of different language processing systems, and discusses how they fare in meeting these constraints. These constraints are:

- Functional independence of syntactic and semantic processing.
- Interaction between lexical and pragmatic processing.
- Multiple access of word meanings.
- Recovery from erroneous inferences.

Most of the systems presented have been offered at one time or another as models of the human language processor, but some have not. These latter systems are included because they occasionally provide a more accurate account of the psychological data than their cognitively-motivated counterparts.

2.2 Independence of syntax and semantics

Chomsky (1957) has noted that people are able to judge the syntactic correctness of sentences that are semantically anomalous, such as:

Colorless green ideas sleep furiously.

Conversely, both Charniak and Winograd have noted that people can assign meaning to agrammatical strings of semantically related words, such as:

Skid crash hospital. (Winograd, 1973)

Fire match arson hotel. (Charniak, 1983)

Given these simple observations, it would be difficult to argue against the proposition that syntactic and semantic processing *can* function independently of each other. This is not to say that these processes *must* function independently or that they never interact. Both processes contribute to an accurate interpretation of the input, but if one process is unable to perform, the other proceeds on its own. Despite the ongoing debate over the degree of interaction between syntax and semantics (e.g., Garfield, 1987), an issue which will be avoided here as much as possible, the previous examples strongly suggest that neither process need be dependent on the other.

Additional support for this proposition comes from the behavior of people with language deficits called aphasias. Broca's aphasics, who have sustained damage to an area in the frontal lobe of the left hemisphere of the brain (Broca's Area), demonstrate an inability to produce a syntactically correct string of words. Wernicke's aphasics, who suffer from damage in the temporal lobe of the left hemisphere, are able to put together long sequences of words in many syntactic constructions, but the sequences have no informational content (i.e., no semantic value). Because deficits in overt speech behavior are most immediately obvious, studies of language use in aphasics have focused on production, but a survey of the literature on language comprehension in aphasics by Caramazza and Berndt (1978) finds that Broca's and Wernicke's aphasics exhibit corresponding deficits in comprehension as well. Caramazza and Berndt conclude that the studies they reviewed support the "functional and neurological independence of syntactic and semantic processing in sentence comprehension" (p. 916). Caramazza and Berndt also say the studies show that "although these processes interact with other cognitive operations (e.g., memory), they can be selectively affected by brain damage" (p. 916). In other words, while there may be interaction between processes, they can be forced to function independently.

Despite this evidence, many computational models of language understanding seem to ignore the simple but important constraint of functional independence of syntax and semantics. Many NLU systems, such as LUNAR (Woods, 1970), SHRDLU

(Winograd, 1973), and PARSIFAL (Marcus, 1980), can be described as *syntax-first* models: the system first performs a syntactic parse, followed by a semantic analysis. While these systems separated syntactic and semantic processing, semantic analysis was dependent upon a correct syntactic parse.[2]

Another group of models are the *conceptually-based* language understanders such as SAM (Cullingford, 1978), PAM (Wilensky, 1978), Ms. Malaprop (Charniak, 1978), and ARTHUR (Granger, 1980a). These systems also employed a two-stage approach to text processing. The first stage was a conceptual analyzer that converted the input text into an intermediate representation based on low-level semantic (i.e., lexical) and syntactic knowledge stored as word definitions. The intermediate representation was then processed by a second stage using pragmatic knowledge structures to fill out the final representation of the input. The theory behind these conceptually-based front ends such as ELI (Riesbeck, 1975) and CA (Birnbaum & Selfridge, 1981) is that "a separate syntactic analysis phase is unnecessary in language understanding" (Birnbaum & Selfridge, 1981, p. 325). Syntactic knowledge is used in these systems, but it is subservient to semantic knowledge. While systems like SAM and PAM demonstrate that some texts can be effectively processed under this assumption, these systems are unable to determine syntactic correctness without a viable semantic evaluation of the input. Humans, on the other hand, demonstrate no such lack of ability.[3]

The assumption of no separate syntactic analysis is carried on in the *integrated* understanding models such as IPP (Lebowitz, 1980) and BORIS (Dyer, 1983). In

[2]SHRDLU actually interleaved syntactic and semantic processing, but the dependence of one process on another was there nevertheless.

[3]Small and Rieger's (1982) Word Expert Parser and Wilks's (1975b) Preference Semantics also make similar assumptions about combining syntactic and low-level semantic knowledge in word definitions, although they do not address the role of pragmatic knowledge in language understanding. Lytinen (1984) and Marcus (1984) present additional arguments against such assumptions.

these systems, the two stages are fused into one process, and lexical and pragmatic analysis is performed incrementally as the individual words of the input text are read. Still, syntax plays a subservient role in these models.

A different realization of the "no separate syntax" assumption is seen in an integrated model of human language understanding called DMAP (Riesbeck & Martin, 1986). DMAP (Direct Memory Access Parsing) uses marker-passing to search an associative network for semantic relations between the words of an input text. Unlike ATLAST, however, DMAP's syntactic knowledge is embedded in its semantic network. Thus, marker-passing in DMAP finds syntactic as well as semantic relationships, and the distinction between syntactic correctness and semantic validity is lost. Riesbeck and Martin acknowledge that DMAP "cannot do something [they] think is very important for it to do, namely, recognize conceptual structures given a few scattered key clues" (p. 223), but they offer no solution to the problem.

Another NLU system which uses marker-passing in a semantic network but departs from the integrated understanding paradigm is Hirst's ABSITY (1988a, 1988b). ABSITY (A Better Semantic Interpreter Than Yours) is a semantic interpreter which employs an individual process or demon attached to each word of a sentence to find that word's correct meaning and case. Ambiguity resolution is aided by selectional restrictions, syntactic cues, or the discovery of associations with other words in the sentence via marker-passing. ABSITY runs concurrently with a PARSIFAL-like syntactic parser (Marcus, 1980) and Hirst's design enforces modularity of the respective knowledge sources. ABSITY diverges from our set of constraints, however, in that it is dependent upon its parser for input and cannot make an independent semantic evaluation of an input string.

An attempt to merge conceptually-based understanding with modularity of syntactic and semantic knowledge sources is offered by MOPTRANS (Lytinen, 1984; 1986). MOPTRANS uses an autonomous set of rules that specify how sequences of syntactic constituents can be attached to each other. However, while the knowledge sources are independent, once again the processing is not. Syntactic rules are applied only "if the syntactic attachments that they make are judged by the parser's semantic analyzer to be semantically appropriate." (Lytinen, 1986, p. 576). MOPTRANS finds possible semantic connections between concepts, chooses the best one, then looks at the syntactic knowledge base for a rule that supports this choice. If no syntactic rule supports the choice, that choice is removed from the list of possible connections, and the process repeats. This sequence continues until no possible semantic connections remain. At this point, a syntactic rule for making syntactic attachments is selected by making an informed guess. Consequently, despite the modularity of knowledge sources, there is a processing dependency established: semantic connections cannot be inferred if no syntactic rule exists to support them. What happens when MOPTRANS is presented with the agrammatical but conceptually-related string of words? It cannot make sense of it syntactically, which is desirable, nor can it make sense of it semantically, which is undesirable since humans do appear to make those inferences despite the lack of syntactic correctness.[4] To be fair, Lytinen makes no claims about MOPTRANS's relevance as a cognitive model of human language understanding. Yet from a cognitive perspective, MOPTRANS is interesting because it addresses some of the problems of the conceptually-based understanders.

[4]Lytinen (1984) briefly describes a solution to this problem, inspired by Charniak's PARAGRAM parser (Charniak, 1981), in which the satisfaction constraints on syntactic rule selection are relaxed. The choice of a syntactic rule then is no longer based on the result of a yes/no test, but the degree to which one rule supports the proposed semantic connection better than the other rules.

2.3 Lexical and pragmatic processes interact

A second constraint suggested by experimental evidence is that lexical inference decisions immediately influence pragmatic inference decisions, and that pragmatic decisions in turn immediately influence lexical ones. Researchers would appear to be in general agreement that there is a dependent relationship between the two levels of processing: most if not all models of language understanding that deal with pragmatic inference decisions base those decisions either directly or indirectly on decisions at the lexical level. In fact, it is difficult to imagine a useful model of language understanding which did not make that assumption. However, many models of language understanding have ignored the effect that the choice of higher-level knowledge structures has on word sense disambiguation.

Another aspect of the interaction between lexical and pragmatic inference processing is that the effects of decisions at one level are immediately felt at the other level. A more general description of this phenomenon is offered by Just and Carpenter (1980). They call it the *immediacy assumption*:

> ...a reader tries to interpret each content word of a text as it is encountered, even at the expense of making guesses that sometimes turn out to be wrong. Interpretation refers to processing at several levels such as encoding the word, choosing one meaning of it, assigning it to its referent, and determining its status in the sentence and in the discourse. The immediacy assumption posits that the interpretations at all levels of processing are not deferred; they occur as soon as possible.... (p. 330)

The immediacy assumption is supported by investigations of eye movements of human readers, as is the assumption of pragmatic influence on lexical decisions. These studies show that a reader's eyes will fixate longer on ambiguous words than on unambiguous words (Carpenter & Daneman, 1981), that a reader's eyes will regress or backtrack immediately when the reader encounters text that contradicts a previous lexical decision instead of continuing through the text (Carpenter & Daneman, 1981), and that gaze duration will increase on a word that can be integrated into the context established by a previously read sentence (Just & Carpenter, 1978). The increased gaze duration on an ambiguous word suggests that the language processor has stopped to resolve the ambiguity before processing further text. Similarly, the regression caused by the contradiction and the increased fixation on a word that can be integrated into the existing context both support the contention that the human language processor tries to complete current processing tasks before moving on to new ones.

The strength of this supporting evidence is questionable, however, as it rests on another assumption, the *eye-mind assumption.* According to Just and Carpenter (1980), the eye-mind assumption is that the gaze duration on a newly fixated word is directly proportional to the amount of time required to process that word. Some researchers argue that the connection between eye movements and language processes is less direct than Just and Carpenter suggest: the relationship between eye movements and language processes may be mediated by the filling and emptying of an input buffer (Shebilske & Fisher, 1983).

Further independent support for both the immediacy assumption and the eye-mind assumption comes from a number of studies of lexical access and disambiguation which indicate that reading an ambiguous word triggers the retrieval of all meanings of that word, regardless of the context, after which the meaning most appropriate to the context is selected very quickly, possibly within 200 milliseconds of reading the ambiguous word (Lucas, 1983; Seidenberg, Tanenhaus, Leiman, & Bienkowski, 1982; Simpson, 1981; Tanenhaus, Leiman, & Seidenberg, 1979).

The immediacy assumption implies that human language understanding is performed in one pass of the input. On the other hand, the two-stage models, whether syntax-first or conceptually-based, require the equivalent of two passes of the text. With the syntax-first models, the first pass is a syntactic parse and the second pass is a semantic analysis. In the two-stage conceptually-based models, the two passes correspond to lexical inference processing in the first stage and pragmatic inference processing in the second stage, as discussed above. Builders of integrated understanding models such as IPP and BORIS have successfully incorporated the immediacy assumption but, as noted above and will be discussed further below, the integrated models suffer from other problems.

2.4 Multiple access of word meanings

In his comprehensive survey of research regarding the processing of ambiguous words, Simpson (1984) states that the main issue in this research has been the nature of interaction between the component processes of word recognition: does context directly influence lexical access or does it influence a postaccess decision process? Accordingly, approaches to lexical access and disambiguation can be divided into two classes: *selective-access* models and *exhaustive-access* models.

2.4.1 Selective-access models

Selective-access models follow the premise that the existing context predetermines to some extent which meanings of an ambiguous word will be retrieved when that word is processed. Some psychological models use a terminating ordered search in which a list of the meanings of a word is examined serially in an order determined by their frequencies of use (Forster, 1976; Hogaboam & Perfetti, 1975). That is, the most frequently used meaning is recalled first and evaluated against the context. If this meaning is appropriate, the search terminates; otherwise the next most frequently used meaning is recalled and evaluated, and so on until an appropriate meaning is found. Simpson calls these *ordered-access* models. Another type of selective-access model is exemplified by Morton's *logogen* model (Morton, 1969; 1979). In this model, each meaning of an ambiguous word is represented by a collection of attributes called a logogen. Each logogen is sensitive to contextual influence in that it may be primed indirectly by the processing of previous words. When the ambiguous word is processed, only those meanings whose logogens exhibit a greater degree of priming will be recalled. Simpson refers to this type of model as a *context-dependent* model.

Many AI models have adopted the selective-access approach. For example, each meaning of an ambiguous word in Riesbeck's ELI is represented by one or more selectional restrictions called requests (Riesbeck, 1975; see also Birnbaum & Selfridge, 1981). These requests check the local context for the existence of specific words or phrases and are executed in sequential order. If the requests for one meaning are sufficiently satisfied, all requests representing competing meanings are de-activated, including those not yet executed. Small and Rieger's Word Expert Parser uses a similar technique (Small & Rieger, 1982). Wilks uses a relaxed form of selectional restrictions in his Preference Semantics model (Wilks, 1975a). In this model, a word meaning is represented by a set of semantic preferences. Semantic preferences are like selectional restrictions in that they test for specific attributes in the local context, but they need not be completely satisfied. When an ambiguous word is processed, the meaning with the greatest number of satisfied preferences is chosen.

Other AI models rely on a more global context, employing contextual or scriptal lexica in which special-purpose dictionaries are associated with pragmatic knowledge structures (e.g., Charniak, 1981; Cullingford, 1978). Each lexicon contains only the contextually appropriate meaning for ambiguous words that might be encountered while processing language with that script or other pragmatic knowledge structure. When a word is encountered by an understander, the lexica associated with the active scripts are examined for the word and its appropriate meaning. If an appropriate meaning is not found in the scriptal lexica, the search proceeds to a default lexicon which contains only standard definitions.

2.4.2 Exhaustive-access models

The variations of the selective-access model suffer from the limitation that in some situations the most appropriate meaning of an ambiguous word is never considered. The ordered-access model incorrectly assumes that the most frequently used context-appropriate meaning will always be the correct meaning. Birnbaum (1985) points out that the assumptions of the context-dependent model are similarly misguided. An ambiguous word used within a script will not always refer to the predetermined script-specific sense of the word, and restrictions attached to individual words select the appropriate meaning in some cases but they inevitably lead to the selection of inappropriate meanings in other cases. This limitation is overcome if *all* meanings of an ambiguous word are considered, as in the exhaustive-access model.

With the exhaustive-access model, all meanings of an ambiguous word are activated regardless of context, after which the meaning most consistent with the context is selected. This model is strongly supported by the results of recent experiments (Onifer & Swinney, 1981; Seidenberg et al., 1982; Swinney, 1979; Tanenhaus et al., 1979). There is evidence that this exhaustive-access process is sensitive to the relative frequency of the word meanings (Lucas, 1983; Simpson & Burgess, 1985) and in some cases is influenced by local context (Seidenberg et al., 1982), so the ordered-access and context-dependent models are not unfounded. Still, the exhaustive-access model appears to be the best explanation of human lexical processing offered to date and has been incorporated in a few computational models of language understanding.

Charniak (1983) proposed a model of language understanding that adheres to the constraint of multiple access of word meanings. This model consists in part of a marker-passer that searches for explanatory connections in a relational memory, and a syntactic analyzer that runs in parallel with the marker-passer. These two modules feed a semantic analyzer that constructs a representation of the input with the help of a "path-checker" that evaluates the connections found by the marker-passer.

Information about objects and actions is stored in memory as predicates. Any given predicate can pass markers to other predicates that represent objects and actions directly related to the first predicate. The marker-passing scheme provides for multiple access of word meanings in that markers are passed to the predicates associated with all senses of a given word. These word sense suggestions are sent to the semantic analyzer at the same time that the syntactic analyzer is sending it structural information. The semantic analyzer, assisted by the path-checker, uses this information to make final decisions about word meanings.

In separating syntactic analysis from marker-passing, Charniak imposes a relationship of functional independence between syntactic and semantic processing, thereby meeting another of the constraints presented earlier in this chapter. The architecture clearly allows for the successful processing of an agrammatical string of semantically related words such as "fire match arson hotel." Charniak does not directly address the issue of semantically anomalous sentences such as "Colorless green ideas sleep furiously," but there is nothing in Charniak's proposed architecture to suggest that such strings could not be processed. In addition, Charniak's proposal meets the constraint of interaction of lexical and pragmatic processing in that lexical and pragmatic information are represented in a single relational memory and all relevant inference decisions are made by the combined action of the semantic analyzer and path-checker.

As Charniak's theory evolved into a working program (Charniak, 1986), his proposed architecture had changed somewhat, and the system is now called Wimp (Wholly integrated marker passer). He describes the path checker as a resolution-based theorem prover. A path, which consists of terms for nodes and first order predicate calculus formulas for links, is now considered to be the basis of a proof that the terms or words at either end of the path actually exist in the story being processed. The only resolvents used by the theorem prover are those contained in paths found by the marker-passer. This constrains the search for resolvents and avoids the combinatorial explosion which is typically associated with resolution-based theorem provers. All semantic processing is done by the marker-passer and the path checker, and the separate component which built a semantic representation of the text in the earlier version of the theory is now eliminated. The syntactic component is still functionally independent of the semantic processes, so Wimp still conforms to that constraint as well as those of lexical/pragmatic interaction and multiple access of word meanings.

In describing Wimp, Charniak professed some doubt about how well a language understanding system based on marker-passing would perform with a much larger knowledge base (Charniak, 1986). Accordingly, Wimp's successor, Wimp2, has no marker-passing component; all semantic processing is driven by a formal logic (Charniak & Goldman, 1988). Another evolutionary change is that the semantic processor is now dependent upon parse trees provided by the syntactic processor. Consequently, Wimp2 looks very much like the traditional syntax-first language understanding systems whose faults inspired the original Wimp theory (Charniak, 1983). While improving performance and simplifying the representation, Charniak appears to have sacrificed cognitive modeling accuracy.

Pollack and Waltz's connectionist model of language understanding (Pollack, 1987; Pollack & Waltz, 1982; Waltz & Pollack, 1984; 1985) also employs spreading activation but does not use the marker-passing style found in ATLAST, ABSITY, DMAP, or Wimp. Instead, Pollack and Waltz's model employs a quantitative spreading activation network which works through the iterative adjustment of the strength of activation at the individual nodes. The network is similar to those described in detail by Feldman (1981) and McClelland and Rumelhart (1981; Rumelhart & McClelland, 1982). In Pollack and Waltz's model, a given set of nodes in memory and the connections between them can represent all or part of an interpretation of some input text; the same holds true for the marker-passing models discussed earlier. What distinguishes Pollack and Waltz's approach from more traditional symbolic-processing models is that their model chooses correct interpretations strictly through the iterative adjustment of activation strengths, without the use of a separate rule-based decision-making process.

Pollack and Waltz's model operates by first constructing a network of weighted nodes and links from an input sentence. This network is divided into four interconnected levels, with each level representing a modular linguistic knowledge source.

The first level is the input level, which is nothing more than the words of the input text. The input level is connected via activation links to the lexical level which consists of clusters of meanings and syntactic categories for each of the words. Within a meaning cluster, there are mutual inhibition links between all the meanings, and some combination of activation and inhibition links between the meanings and the lexical categories.

The lexical level is connected by activation links to two other levels. One of these levels, the syntactic level, is a subnetwork of the possible syntactic parses of the input text. This network, which is constructed by a chart parser which preprocesses the input text, has activation links between phrase markers and their constituents, and inhibition links between pairs of phrases that have common constituents. The other level is the contextual level, which is made up of case frames whose purpose is to direct word sense selection. The syntactic and contextual levels are not directly connected to each other; they are only connected indirectly through the lexical level.

Once the network has been constructed, the iterative weight adjustment process is applied. Words at the input level are activated in the order that they appear in the input stream, and activation begins to spread throughout the network. Over several iterations, a subnetwork of well-connected nodes across different levels will emerge. Those nodes which are negatively connected to this subnetwork will be suppressed. Putting inhibitory links between nodes which represent, for example, well-formed phrases with mutually-exclusive shared constituents ensures that only one of the possibilities will survive. This same principle is used to disambiguate between competing lexical categories for the same word, competing word senses, and conflicting thematic role assignments.

From a broader perspective, Pollack and Waltz's model is a system "in which the knowledge sources are modular, but the processing is fully integrated" (Pollack and Waltz, 1982, p. 50).[5] This scheme effectively gives equal importance to all sources of language knowledge.

Another connectionist model of language processing is offered by Cottrell and Small (1983; Cottrell, 1985). Their model attempts to form some correspondence between the elements of a theory of language processing based on psychological data and the mechanisms involved in carrying out those processes. Cottrell and Small's model is strongly influenced by the experimental evidence for multiple access of word meanings. In addition, their model is further influenced by the data on language processing deficits in aphasics supporting modularity, and they suggest that an accurate model of human language comprehension should be "lesionable"; in other words, the model should exhibit similar aphasic behavior if the appropriate linguistic module is artificially damaged in some way.

Cottrell and Small's model is an active semantic network scheme based on the work of McClelland and Rumelhart, as is the Pollack and Waltz model. Their network is also divided into four components. The lexical level is comprised of a unit for every word in the language. The lexical level is connected to the word sense level which consists of meaning nodes for the words in the lexical level. Different noun meanings for the same word inhibit each other, as do different verb meanings for the same word, but there are no inhibitory connections between competing noun and verb meanings. Nominal concepts and verbal concepts are stored in separate subnetworks at the word sense level, based in part on differences between nouns and verbs reported by

[5]Some readers may dispute the claim of full integration of processing because of Pollack and Waltz's use of a chart parser in a stage prior to the iterative weight adjustment stage. Note that once the network is constructed, however, the various syntactic alternatives are evaluated at the same time as the competing semantic interpretations.

Gentner (1981). In addition, function or closed class words are maintained in a third subnetwork at this level because, say Cottrell and Small, Broca's aphasics usually show an inability to use functor information (though this may not account entirely for their comprehension deficits, they say). This behavior indicates that function words are in a class distinct from nouns and verbs.

The case level consists of nodes representing the possible relationships between the predicates and objects. This representation uses several hundred roles that are more specific than agent and object, for example, but fall into those classes. (This representation was inspired by Fahlman (1979).) These nodes are then connected to fewer word senses than agent and object would be, and carry much more information directly. The numerous thematic relations at this level help to build specific expectations for role assignments. These specific expectations help to resolve lexical ambiguities.

The syntax level is connected to both the word sense and case levels in order to constrain which bindings may be made based on sentence structure. Cottrell and Small state that they have given the syntax level less attention than the other levels.

As lexical items are activated in sequence, so are their word senses. The word senses in turn activate the nodes at the case level. The relation which best fits the input will be a stable group of connected nodes in which the overall excitation exceeds the overall inhibition. Cottrell and Small acknowledge that higher levels in the network are needed for making general inferences and for long term memory but leave the work on these levels for the future. In the meantime, these contextual knowledge sources are simulated by pre-biasing the network.

Of all the language understanding models reviewed so far, only the three discussed immediately above satisfy the constraints of functional independence of syntax and semantics, interaction between lexical and pragmatic processing, and multiple access of word meanings. On the other hand, none of these models addresses the problem of error recovery. The following section describes the few models of language understanding that have been concerned with error recovery.

2.5 Recovery from erroneous inferences

Researchers in natural language understanding have often assumed that although text may be ambiguous, it is not misleading. This assumption is usually made for the sake of convenience, not because of belief. Nevertheless, the question of how a text understander can correct its mistakes in interpretation is seldom asked. Of course, text is often misleading, either intentionally or accidentally, and humans frequently appear to be able to recover from mistakes in interpretation quite gracefully. One well-known example of misleading text is due to Lashley (1951), who used the following spoken sentence to illustrate the shortcomings of the behaviorist school of psychology:

> Rapid righting with his uninjured hand saved from loss the contents of the capsized canoe.

People will often hear the second word as "writing" and realize their mistake only when they hear "the capsized canoe." Lashley argued that the behaviorist explanation of simple associative chains between a stimulus and a response failed to account for this serially ordered behavior (Gardner, 1985).

Sadly, cognitive science has devoted little effort to finding an explanation of how the understander detects the error or what the understander does once it is detected. Artificial intelligence researchers occasionally mention the need for an ability to recover from errors in their models but then postpone the issue as a topic for further research (e.g., Birnbaum & Selfridge, 1981; Lebowitz, 1980; Lytinen, 1984). Psycholinguistics is similarly marked by a scarcity of investigation into error recovery processes during text understanding. There are, fortunately, some exceptions in each field.

One exception is McDermott's TOPLE (1974). TOPLE is a computer program which understands simple declarative sentences describing the action of a monkey in a room with objects such as a ball, a table, or a bunch of bananas. TOPLE does not see the actual English sentences; instead, its input is a set of semantic propositions in a modified predicate calculus formalism which represent the output of a natural language front-end. As TOPLE processes the propositions, it makes plausible inferences and adds them, in the form of assertions, to its limited model of the world. When TOPLE encounters an ambiguity, it builds multiple models of the world to be considered simultaneously. A decision is made when TOPLE finds confirming evidence for one of its competing world models. At that time the alternate models are discarded and are no longer accessible to the system.

For example, assume TOPLE knows that there is a monkey and a table in a room, and that a ball and a banana are under the table. If TOPLE is told that the monkey went to the table, TOPLE will build two models of the future: one in which the monkey plays with the ball, and another in which the monkey eats the banana. (TOPLE assumes that the monkey has goals such as wanting to play and wanting to eat.) If TOPLE is then told that the monkey picked up the banana, TOPLE will discard the model in which the monkey plays with the ball.

TOPLE's predictions about future events are limited by what it assumes to be true about its limited world. If TOPLE knows only that the room contains only a monkey and a bunch of bananas suspended from the ceiling out of the monkey's reach, TOPLE will make the predictive inference that the monkey will jump up, grab the bananas, fall back to the floor and eat them. This seems perfectly reasonable. What seems unreasonable is what happens when TOPLE is then told that there is also a box in the room. TOPLE now predicts that the monkey will go to the box, move the box to the spot beneath the bananas, climb the box, grab the bananas and eat them. Based on the example with the monkey, ball, and banana, one might well expect that TOPLE will now have two competing models of the future. What TOPLE actually does, however, is adopt only the model in which the monkey uses the box; TOPLE discards the model in which the monkey jumps for the bananas. TOPLE does this because it assumes that objects added to its model of the world are to be used by the monkey, so it no longer believes the prediction that does not include the box. McDermott does not say what will happen if TOPLE is now told that the monkey jumps for the bananas, but presumably TOPLE will regenerate the world model that it has just discarded and discard the model in which the monkey uses the box. If TOPLE had retained both models and waited to see which was confirmed, it could have saved itself some work.

TOPLE's inferential and error recovery abilities are limited to predictions about future states and events. It cannot undo incorrect inferences generated from propositions that it processed well before the one it has just read. One model which can correct old erroneous inferences is Granger's ARTHUR (1980a, 1980b). ARTHUR follows from SAM (Cullingford, 1978) and PAM (Wilensky, 1978), but is designed to understand the misleading stories which SAM and PAM could not. ARTHUR generates tentative initial inferences using pragmatic knowledge structures (i.e., scripts and plans) and then re-evaluates those inferences in light of subsequent story information by applying evaluation metrics. The re-evaluation is made possible because ARTHUR maintains an inference-fate graph which contains *all* the plausible inferences generated during story processing, whether or not they appear in the final representation, along with information about the current status of each inference. ARTHUR is able to revise both tentative inferences about future events and accepted inferences about previously processed events without reprocessing the story or recomputing discarded inferences because it retains all possible interpretations of the text.

This approach to error recovery has a drawback, however: when ARTHUR processes increasingly longer stories, the number of plausible inferences to be maintained, the memory required to maintain them, and the bookkeeping required to keep track of them becomes prohibitive. A possible solution to this problem is to limit the number of plausible but unused inferences to be retained, as in an early version of Norvig's FAUSTUS (1983). FAUSTUS uses a marker-passing system to locate relevant pragmatic memory structures, called frames, to explain an input text. A frame can be either inactive, active and currently used as an explanation for the text, or previously active and not currently used as an explanation. These previously-active frames are temporarily stored in a separate data base in case a currently-active frame proves to be a poor explanation. In this event, a competing, previously-active frame can be reactivated to supplant the incorrect initial choice. If a previously-active frame is not reactivated, it quickly reverts to an inactive state. The method of error recovery used in this initial version of FAUSTUS is very much like that implemented in ATLAST. As FAUSTUS has evolved, however, its error recovery capability has not, and FAUSTUS no longer has the ability to recover from erroneous inferences (Norvig, 1987).

Another method for correcting old erroneous inferences is to use dependency-directed backtracking in a non-monotonic reasoning system, as O'Rorke (1983) proposed for a model called RESUND. This method was later implemented in a system called WATSON (Orejel-Opisso, 1984). The idea behind dependency-directed backtracking is that when a new belief or assertion is inferred from old ones, the understander records that the new assertion is made only because the old assertions are assumed to be true. If one or more of the old assertions are later proven to be false, any newer assertions that depend on the presumed truth of the old assertions must be retracted, as must be the old, false assertions. In this way, the understander is able to undo old erroneous inferences.

There are some appealing features of a language understanding system based on dependency-directed backtracking. One such feature is that the system can explain its reasoning; the dependency relations enable the system to justify any assertion in its knowledge base by describing the chain of inferences that led to the assertion in question. Another attractive feature is that the dependency relations can be maintained indefinitely, offering the potential for error recovery regardless of the length of input processed or the amount of time passed since the erroneous inference was made. These features, though, are not without their costs.

The disadvantage of dependency-directed backtracking in a model of human language understanding is that it makes the understander perform too well. Imagine a reader who makes an erroneous inference while reading the first paragraph of a short mystery story but finds nothing to contradict the inference until many paragraphs later. A computer program such as RESUND or WATSON using dependency-directed backtracking should have no problem in correcting its mistake and revising its interpretation of the story. A human reader, on the other hand, probably will be confused and possibly will hunt through previously-read text to find the source of the conflict. The human reader might in fact be using dependency relations of some sort in constructing an interpretation of the story. Yet it is unlikely that the human reader is able to maintain a very large number of such relations for an indefinite time without substantial degradation in his or her ability to retrieve and revise those relations. Systems using dependency-directed backtracking might be made to perform more like human understanders through the application of constraints on the number or duration of maintained relations; constraints along these lines are employed by both FAUSTUS and ATLAST.

TOPLE, ARTHUR, FAUSTUS, and RESUND represent four different methods for recovering from erroneous pragmatic inferences, all utilizing a two-stage architecture. Because of the two-stage architecture, however, these systems violate the immediacy assumption: each requires complete propositions before any pragmatic processing begins, yet the immediacy assumption asserts that inferencing in humans is done with incomplete information, as in the case of Lashley's example above.

One model which does adhere to the immediacy assumption and also appears to address the problem of error recovery is proposed by Thibadeau, Just, and Carpenter (1982). This model, called READER, is based on the eye-movement studies cited previously as experimental evidence supporting the immediacy assumption (Just & Carpenter, 1980). READER is a model of human text understanding whose processing cycles bear some correspondence to human gaze durations on the individual words of the text. READER tries to interpret each successive word as soon as possible by applying separate but communicating processes at several levels of analysis without relying entirely on any single level. READER's memory is a semantic network in which relationships between concepts from the text are found by *directed activation* as opposed to spreading activation. With directed activation, both the depth and breadth of propagation is controlled by READER's procedural knowledge about language. In other words, directed activation means that the analytical processes tell READER which nodes it should activate at any given time, while spreading activation propagates outward in all directions until some limit is reached.

It was stated above that READER *appears* to address the error recovery problem. Thibadeau et al. say that the principle of immediacy causes READER to choose meanings which are occasionally contradicted by later text, but that their model "has other heuristics that make error recovery relatively straightforward" (Thibadeau et al., 1982, p. 180). Unfortunately, they do not provide any details about READER's error recovery heuristics, but they do state that their approach to lexical disambiguation is influenced by the eye fixation studies of Carpenter and Daneman (1981) which, as noted earlier, show that a reader's eyes will backtrack when he or she reads text contradicting a previous lexical decision.[6] Assuming that READER uses backtracking and reprocessing in these situations, it can be said that the model meets three of the four constraints: functional independence of syntax and semantics, interaction of lexical and pragmatic processing, and error recovery. READER fails to meet the constraint of multiple access of word meanings, however: READER's lexical access process is similar to Morton's (1969) logogen model in which the meaning of an ambiguous word is chosen through the accumulation of contextual (both syntactic and semantic) information. When the logogen has accumulated a specified amount of

[6]The implications of Carpenter and Daneman's findings for ATLAST are discussed in Chapter 4.

information (i.e., reached a threshold), the associated word meaning is made available to other linguistic processes. The logogen model therefore represents a form of selective access in which context predetermines the meanings that will be considered.

Although there is little experimental data on recovery from incorrect decisions during text understanding in human subjects, what data there is suggests that error recovery is an integral part of language understanding, not something to be added on to existing models as an afterthought. Providing an adequate explanation of human error recovery behavior places a constraint on the architecture of a computational model of language understanding that few existing models would be able to meet without drastic revision.

2.6 Conclusion

Different constraints can be applied in the development and evaluation of any model. The constraints presented herein are important because they cut across the traditional partitionings of lexical, syntactic, and pragmatic inference processes. Those models meeting more of these constraints offer more plausible explanations of human language processing than those which meet fewer constraints. This chapter has discussed how some of these models fare in meeting the important constraints. (This discussion is summarized in Table 2.1.) As will be demonstrated in the chapters to follow, ATLAST is the first model to meet all these constraints.

System	functional independence of syntactic and semantic processing	interaction between lexical and pragmatic knowledge	exhaustive access of word meanings	recovery from erroneous inferences
Quillian, 1969 (TLC)	no	yes	yes	no
Woods, 1970 (LUNAR)	no	no	no	no
Winograd, 1973 (SHRDLU)	no	no	no	no
McDermott, 1974 (TOPLE)	no	no	no	yes
Riesbeck, 1975 (ELI)	no	no	no	no
Wilks, 1975 (Preference Semantics)	no	no	no	no
Charniak, 1978 (Ms. Malaprop)	no	no	no	no
Cullingford, 1978 (SAM)	no	no	no	no
Wilensky, 1978 (PAM)	no	no	no	no
Granger, 1980 (ARTHUR)	no	no	no	yes
Lebowitz, 1980 (IPP)	no	yes	no	no
Marcus, 1980 (PARSIFAL)	no	no	no	no
Birnbaum & Selfridge, 1981 (CA)	no	no	no	no
Pollack & Waltz, 1982	yes	yes	yes	no
Small & Rieger, 1982 (Word Expert Parser)	no	no	no	no
Thibadeau, Just, & Carpenter, 1982 (READER)	yes	yes	no	yes
Charniak, 1983	yes	yes	yes	no
Cottrell & Small, 1983	yes	yes	yes	no
Dyer, 1983 (BORIS)	no	yes	no	no
Norvig, 1983 (early FAUSTUS)	no	no	no	yes
O'Rorke, 1983 (RESUND)	no	no	no	yes
Hirst, 1984 (ABSITY)	no	no	yes	no
Lytinen, 1984 (MOPTRANS)	no	yes	no	no
Orejel-Opisso, 1984 (WATSON)	no	no	no	yes
Riesbeck & Martin, 1985 (DMAP)	no	yes	yes	no
Charniak, 1986 (Wimp)	yes	yes	yes	no
Norvig, 1987 (late FAUSTUS)	no	no	no	no
Charniak & Goldman, 1988 (Wimp2)	no	yes	yes	no

Table 2.1: Representative NLU systems and how they fare in meeting the four constraints on understanding.

Chapter 3

How ATLAST Works

3.1 Constraints revisited

The previous chapter introduced four important constraints on computational models of natural language understanding drawn from the psycholinguistic literature:

- Functional independence of syntactic and semantic processing.
- Interaction between lexical and pragmatic processing.
- Multiple access of word meanings.
- Recovery from erroneous inferences.

The importance of these four constraints rests in how they guide the development of a theory of language understanding in humans and the construction of the corresponding model. As a theory is pushed to explain a greater diversity of phenomena, the number of ways in which that theory may take shape is narrowed. This is illustrated in the previous chapter: just the simple observation that people seem to make some sense of syntactically anomalous strings of words casts doubt on the psychological validity of many integrated understanders and thereby eliminates one set of potential answers to the question, "How does the human language processor work?"

The constraints also provide a context within which to evaluate this work. While there is no universal agreement on any of these constraints, more is gained by making strong specific assumptions for one's model and testing them than by making weaker all-encompassing assumptions, inevitably resulting in a nebulous model which is difficult to evaluate and therefore uninformative. This chapter describes in detail the architecture of the ATLAST model, whose design was guided by the constraints listed above.

3.2 Architecture

ATLAST consists of three independent processing components, the capsulizer, the proposer, and the filter, that operate concurrently on a relational memory network. The common goal of these components is to find the parts of memory that best represent the intended meaning of some input text. ATLAST pursues this goal by using marker-passing to search the network for paths that connect senses of open class words from the text. A single path is a chain of nodes and the links which join them. The nodes represent objects, events, or states, and the links correspond to the relationships which may exist between the nodes. Any nodes or links in a path which are not explicitly mentioned in the text are inferred; therefore, these paths are called *inference paths.* A set of inference paths which joins all of the words in the text into a connected graph represents one possible interpretation of the text. In this respect ATLAST resembles a number of other models of text understanding that utilize marker-passing or spreading activation, many of which were discussed in Chapter 2.

The paths that form the current interpretation are called *active paths.* For any given text, however, there may be a great number of possible interpretations, many of which are nonsensical. The problem then is determining which of the possible interpretations provides the best explanation of the text. ATLAST deals with this problem by applying heuristics called *inference evaluation metrics.* These metrics are used to compare two competing inference paths and select the more appropriate one. Two inference paths compete when they connect the same two nodes in the relational network via different combinations of links and nodes. The path that fits better with the current interpretation is activated (i.e., it becomes part of the interpretation). The other path is de-activated but not discarded. Instead, information about that path is retained in order to facilitate error recovery; these paths are called *retained paths.* The choice of one inference path over another is made as soon as ATLAST discovers that the two paths compete.[1] As the marker-passing search mechanism finds more paths, ATLAST constructs an interpretation consisting of those paths which survive the evaluation process. When the marker-passing and evaluation processes end, the surviving active paths make up the final interpretation of the text.[2]

3.2.1 Memory

At the core of ATLAST is a simple memory structure organized as a relational network. The network consists of nodes and links. The nodes represent events, objects, and states; the links represent relationships between the nodes. Many of

[1]An exception to this principle is introduced in Chapter 6.

[2]In theory, the search for inference paths and their evaluation take place simultaneously. In practice, however, ATLAST simulates this concurrency by alternating between marker-passing and path evaluation. During each of these cycles, a new word is read from the input, its meanings are recalled and marked, and all markers in the network are passed a fixed distance. Any path discovered in this way is then examined to see if it competes with an active path in the interpretation as it stands at that time. If so, the evaluation metrics are applied and a choice between the two competing paths is made.

these nodes correspond to meanings of words in ATLAST's vocabulary, and it is from these nodes that the search for inference paths begins.

Unlike other systems which place restrictions on the direction in which links can be traversed (e.g., Riesbeck & Martin, 1986), ATLAST employs two uni-directional links between related nodes, giving the equivalent of a bi-directional link. This allows the search for inference paths to spread from one node to all directly related nodes. Each uni-directional link represents one-half of the relationship between two nodes and is is labeled accordingly. Thus, if there is a link representing the **has-part** relationship connecting one node to another, there is a corresponding **part-of** link in the opposite direction.

The relationships described by the links themselves can be divided roughly into four categories: composition, causality, intentionality, and abstraction. ATLAST's composition relations include **has-part**, **has-instance**, **has-attribute**, and **has-role-filler**. The causality relation is **has-result**, and the intentionality relations are **has-goal**, **has-plan**, **has-planstep**, and **has-precondition**. The one abstraction relation is **viewed-as** (its corresponding relation in the opposite direction is also **viewed-as**). ATLAST's **viewed-as** relation is inspired by the **view** relation in Wilensky's (1984) KODIAK knowledge representation language, although **viewed-as** relates only two concepts while **view** is a three-part relation. For example, in KODIAK, a **view** relation can be used to declare that the concept `SELL` is a `COMMERCIAL-TRANSACTION` viewed as an `ACTION`. If we wanted to describe the same relationship in ATLAST, we might say that a `COMMERCIAL-TRANSACTION` can be **viewed-as** an `ACTION`, and that `SELL` is an **instance-of** a `COMMERCIAL-TRANSACTION`. In comparison, ATLAST's minimal representation language is perhaps less elegant and less precise than KODIAK, but the two representations exist to solve different problems: KODIAK is a robust framework for defining a wide range of semantic relations while ATLAST's representation language is a vehicle to facilitate the demonstration of ATLAST's processes. There is no reason to believe, however, that ATLAST's processes could not be adapted to a more sophisticated representation scheme like KODIAK.

3.2.2 Capsulizer

The capsulizer is essentially a syntactic parser based on a simple augmented transition network (ATN) grammar and is the only process that sees the actual input text. As the capsulizer reads the input text one word at a time, it retrieves the syntactic category information (e.g., noun or verb) associated with that word and activates any word senses associated with that word by placing markers at those word senses. The activated word senses serve as a starting point for the search for inference paths, which is discussed below. The word senses are not used in any decisions made by the capsulizer, although pointers to the word senses are retained. As ATLAST accumulates syntactic category information, it uses the ATN grammar to make decisions about syntactic relationships within the phrases of the input text. These intra-phrasal decisions, along with the pointers to the word senses which comprise the phrases, are packaged and passed along to the filter as capsules of information. In addition, for any input word that has more than one word sense (i.e., a word sense ambiguity), a capsule containing the pointers to the different word senses is sent to the filter, which will eventually select the correct word sense.

3.2.3 Proposer

The proposer suggests possible inference paths that might explain the input text. Essentially, the proposer employs a simple marker-passing process to perform a breadth-first intersection search of the relational network for connections between word senses that have been activated by the capsulizer. The proposer extends the frontiers of the search space by traversing the links leading away from the origins of the search (i.e., the word senses directly marked by the capsulizer) and placing new markers at the nodes it encounters as it goes. This search process is roughly analogous to wavefronts of activation spreading outward from different sources. Each marker carries with it enough information to describe the path that extends from the node where the marker resides to the node that represents the source of the activation. Thus, two markers from different origins arriving at one node represent a potential inference path to be considered for inclusion in ATLAST's interpretation of the text.

How markers are passed

The extent of the search as it radiates outward from a given origin is represented by a set markers. Each marker is uniquely identified and contains information that aids in controlling the search and reporting paths which connect two origins. The information associated with each marker includes:

- **Origin node:** A pointer to the node from which this marker began its search.
- **Parent:** A pointer to the marker which spawned this marker (i.e., the marker that is one link closer to the origin).
- **Host node:** The name of the node marked by this marker.
- **Distance:** This value is roughly equivalent to the number of links that were traversed from the origin in order to place this marker at its host node. The

reason that these values are not exactly equal is that some of the links may be **preferred** links which, as described below, do not incur any cost when traversed.

ATLAST keeps track of the markers that make up the frontier of the search from each origin (i.e., the most recently placed markers). When the proposer is invoked, each of these markers spawns children which are placed at the nodes immediately adjacent to their parent's host node. (A marker does not pass a child marker to its parent's host node as this would only create a cycle to be ignored later.) These child markers are copies of their parent. The child carries the same origin information as its parent, the child's parent and host node information is updated accordingly, and its distance value is its parent's distance value increased by one (unless the link separating the child and parent is a **preferred** link).

Without some means of limiting the spread of markers throughout memory, ATLAST would be overwhelmed by the number of inference paths proposed for evaluation. To avoid this problem, the depth of the proposer's search is controlled through a parameter that imposes a global upper limit on the markers' distance value. If a given marker's distance value has reached the limit, that marker cannot spawn child markers. This method of accounting for the decay of activation is certainly more simplistic than other marker-passing schemes, such as Hendler's SCRAPS (1986) or Anderson's ACT* (1983), in that the markers are not assigned continuously variable activation energies which are depleted by both link traversal and fan-out, and there are no complicated formulas for calculating the decay rate or the cost of link traversal. Instead, ATLAST's distance values can be viewed as integer-valued activation energies that are depleted only by the cost of traversing links. When copies of a marker at a given node are passed to adjacent nodes, each new marker is given an activation energy equal to that of its parent marker less the cost of traversing the link to the adjacent node; the parent's activation energy is not divided among the new markers according to the number of links leading away from the node as it is with other systems.

The limit on the depth of search places an upper bound on the length of the paths that the proposer will find, although some paths may in fact exceed that limit. These exceptions can occur when the path in question contains one or more **preferred** links. While the traversal of most links in ATLAST's memory is assumed to incur some cost, traversal of those links that are designated as **preferred** links does not incur any cost and is not included in determining the extent of the search. The use of **preferred** links will be discussed in Chapter 6.

How paths are proposed

When a marker is placed at a node, the proposer checks to see if there are other markers at the same node. If so, the pairing of the newly-arrived marker with each of the other markers represents a potential inference path between the two origins of each pair of markers. These potential inference paths must meet minimum acceptability requirements before they are considered by the filter's evaluation process. A path that connects an origin to itself will not be passed to the filter for evaluation, nor will any path that contains a cycle, regardless of its origins. A path connecting two different sentences (which ATLAST naively assumes to represent different states or events) must meet more stringent requirements: the path must contain at least one link denoting a causal or intentional relationship between the sentences (cf. Schank & Abelson, 1977), and the path's endpoints must be nodes representing actions or states. These constraints, along with the limit on the spread of marker-passing described above, serve to limit combinatorial growth of the number of paths that could be discovered and evaluated. The combinatorial growth problem is discussed in greater detail in Chapter 7.

3.2.4 Filter

The filter performs two functions; the first is that of inter-phrasal syntax. As capsules are passed from the capsulizer to the filter, the filter makes decisions about the relationships between the phrases represented by the capsules. Inter-phrasal syntax rules enable the filter to make thematic role assignments. The filter's second function is the evaluation of inference paths. When two competing inference paths are proposed, the filter attempts to select the more appropriate path through the application of inference evaluation metrics.

Inter-phrasal syntax

In performing inter-phrasal syntactic analysis, the filter's goal is to construct a framework of causally-connected event descriptions that correspond to the events described by the input text. The framework that is built is very much like the framework that results from parsers using Schank's Conceptual Dependency (CD) representation scheme (Schank, 1975; Schank & Riesbeck, 1981). Like the CD framework, every event in ATLAST's representation consists of an *actor*, an *action*, an *object*, a *direction*, and a *causal* or *intentional relationship* between this event and others. The difference between the two approaches is in how these slots are filled.

Parsers based on CD, such as MARGIE (Riesbeck, 1975), took advantage of an integrated procedural representation of both semantic and syntactic knowledge. Rules attached to word meanings enabled these parsers to construct the correct relationships between the components of an event and between the different events themselves. ATLAST's semantic knowledge is primarily declarative and distributed throughout its memory network, and its syntactic knowledge is a separate set of rules. ATLAST's semantic relationships within and between events are explicitly defined in memory before processing begins. The proposer suggests possible relationships and the filter decides which relationships best represent the input. The syntactic relationships are determined by the syntactic processes of the capsulizer and the filter. Rules in the capsulizer guide the decomposition of the input text into syntactic components. The capsulizer then sends this information to the filter which in turn uses other rules to assign the components to various thematic roles. The filter also looks for cues that alter the normal temporal flow of the text so as to maintain a correct ordering of the events. The result is a framework of pointers leading to nodes that belong to active paths in ATLAST's memory as shown previously in Figure 1.1.

This dissertation concentrates on how ATLAST finds the correct active paths, but the importance of the syntactic component of ATLAST's interpretation, the framework of pointers, should not be underestimated. While the nodes and links of the active paths represent objects, events, states, and some of the functional relationships between them, it is the syntactic component that imposes order on the otherwise nebulous set of active paths. Without this component it would be difficult to know which of the events depicted by the active paths occurred first, or even who did what to whom. This deficit would become more apparent as texts became more structurally complex.

Inference evaluation

As the proposer passes information to the filter about inference paths it has discovered, the filter evaluates each path to determine if that path should be incorporated into ATLAST's changing interpretation of the input text. Because marker-passing is begun at different nodes at different times, as determined by the order of the words in the text, the same path may be discovered many times. It might be argued that this redundancy is computationally inefficient but it is essential to ATLAST's error recovery capability as described in Chapters 4 and 5. Thus, the filter's first task is to see if the proposed path is already part of the current interpretation. If so, no evaluation need be done and the filter can move on to the next proposed path.

On the other hand, if the proposed path is not a component of the interpretation, one of two things may occur. The filter will check to see if the proposed path competes with any path in the current interpretation (i.e., they share the same two origin nodes). If there is no competing path, then the proposed path is the only path found so far that connects the nodes at either end of the path. In this case, the filter adds the proposed path to the interpretation, making this path an active path. Should the filter find a path in the current interpretation that does compete with the proposed path, the filter will apply the inference evaluation metrics to determine which of the two paths fits better with the existing context (i.e., the set of all active paths other than the path competing with the proposed path).

ATLAST employs five different metrics or rules for selecting the more appropriate inference path.[3] All but one of these metrics represent attempts to embody the principle of parsimony. This principle can be defined as explaining the most input

[3]There is also a sixth metric, but its introduction is postponed until Chapter 6.

with the most efficient or economical representation. The concept of representing the most with the least has been incorporated into many models and theories of language understanding through the years, although the word "parsimony" may not have been used to describe it. (See for example the work of Crain and Steedman (1985), Granger (1980b), Kay (1983), McDermott (1974), Quillian (1969), Wilensky (1983), or Wilks (1978).)

The first of the evaluation metrics, the *activation metric*, relies on the assumption that the endpoints of the competing paths are always origins of marker-passing. These nodes are activated or marked directly by the capsulizer as it processes the words from the input text. Any of these nodes may be activated by the capsulizer more than once; in the example of Chapter 5, a node labeled "insect" will be marked twice by the capsulizer if the text contains both the words "bugs" and "roaches." The activation metric favors the path whose endpoints have been activated the greater number of times by the capsulizer, thus giving preference to the path that explains the greater number of different words from the input text.

The *length metric* favors the path that contains fewer links connecting the two origin nodes, thus embodying the concept of using the more concise representation to explain the relationship between two words from the text. The principle of giving preference to the shorter path is assumed by most, if not all, models based on spreading activation.

Inference paths in ATLAST's memory often share nodes with other inference paths. The *reinforcement metric* attempts to make a judgment as to how well each of two competing paths fits with the existing context by counting the number of nodes that each path shares with the other active paths (excluding the two paths being evaluated). The metric prefers the path that has greater reinforcement from, or has more shared nodes with, the other active paths.

Another aspect of the principle of parsimony is to make the most specific interpretation possible of a text. (Wilensky (1983) calls this the Principle of Concretion). ATLAST's memory uses **viewed-as** links to denote relationships of abstraction between two nodes. ATLAST's **viewed-as** link is adapted from the **view** link used in the KODIAK knowledge representation language (Wilensky, 1984). For example, a **viewed-as** link between a node labeled **GOVERNMENT-AGENCY** and a node labeled **GENERIC-EMPLOYER** says that **GENERIC-EMPLOYER** is another way of looking at **GOVERNMENT-AGENCY** or vice versa. Because a path containing a **viewed-as** link describes a more abstract relationship between two nodes than a path not containing such a link, the *specificity metric* favors the path containing fewer **viewed-as** links.

The **viewed-as** link also has a special property in that is has no length as far as the length metric is concerned. The **viewed-as** link does not represent a hierarchical relationship; instead it represents a relation of abstraction and, to some extent, equality, as in "here is a different but perfectly reasonable way to look at this node." Because the two nodes on either end of a **viewed-as** link are in some sense equal, and because the presence of the **viewed-as** node is already measured by the specificity metric, the length metric ignores the **viewed-as** link when computing the length of a path containing that link.

To evaluate competing paths, the filter applies the metrics in the order given above. That is, the filter first uses the activation metric to compare the two paths. If the activation metric favors one path over the other, the evaluation is complete. Otherwise, the filter employs the length metric, followed by the reinforcement metric and the specificity metric. This ordering of the metrics is obviously not the only ordering possible, but it has been determined through trial and error to be the only ordering which enables ATLAST to arrive at correct interpretations for the sample texts on which it has been tested. This should not be construed to mean that the set of metrics is necessarily correct or complete. It is likely that testing ATLAST on additional texts would reveal the need for additional metrics as well as revisions to both the order and the content of the existing metrics, as has been the case so far.

The four metrics listed above do not guarantee the selection of one of the two competing paths; it is possible that the two paths will fare equally well in all four comparisons. In this case a fifth metric, the *no-decision metric*, is invoked. As the name indicates, this rule makes no decision about the competing paths. Instead, it retains both paths, expecting that the processing of more text will change the existing context sufficiently that a decision can be made and that the subsequent evaluation of other paths will remind ATLAST of the tied paths and instigate their re-evaluation. To prevent a situation in which ATLAST is reminded to re-evaluate one tied path and not the other, thus giving the former path a chance to be activated and influence later changes to the active interpretation without considering the latter path, ATLAST records the fact that the two paths are tied. When one of the two tied and retained paths is re-evaluated, the other will be recalled immediately for re-evaluation.

When a decision has been made as to which of the two competing paths provides a better explanation of the input, ATLAST's ongoing interpretation may be altered. If the evaluation metrics selected the path that was already a part of the interpretation, no change is necessary. However, if the previously-active path is not selected by the metrics, it is de-activated and removed from the interpretation, while the selected path is added to the interpretation and becomes an active path. In either case, the less appropriate path is not discarded; instead it becomes a retained path. That is, it is remembered as having been a candidate for inclusion in the interpretation. This is done in order to facilitate recovery from incorrect inference decisions, which is also discussed in great detail in the next two chapters.

3.3 Why marker-passing?

The marker-passing search mechanism described above is a very simple computational metaphor for the concept of spreading activation. Waltz and Pollack (1985) divide spreading activation schemes into two categories: digital spreading activation and analog spreading activation.[4] *Digital spreading activation* includes marker-passing algorithms which perform a breadth-first search for shortest paths in a relational network, while *analog spreading activation* takes place on a weighted network of associations, where "activation energy" is distributed over the network based on some mathematical function of the strength of connections.

This categorization is further supported by Fahlman (cited in Fahlman, Hinton, & Sejnowski, 1983), who divides massively parallel architectures into three classes, two of which correspond to Waltz and Pollack's categories of spreading activation.[5] Fahlman classifies systems by the type of signal that is passed between the processing elements which make up the system: messages, markers, and values. *Message-passing systems* pass arbitrarily complex messages between elements and perform complex operations on those messages. An example of message-passing applied to language

[4]An earlier paper by the same authors (Pollack & Waltz, 1982) makes the same distinction but gives different names to the categories.

[5]A massively parallel architecture is defined by Fahlman, Hinton, and Sejnowski (1983) as a machine with a very large number of (possibly very simple) processing elements working on a single task.

understanding is given by Phillips and Hendler (1981). Because message-passing systems are not necessarily spreading activation systems, they will not be discussed further. *Marker-passing systems*, which pass simple discrete markers between processing elements along dedicated links, correspond to Waltz and Pollack's digital spreading activation systems. Fahlman places his own NETL system (Fahlman, 1979) into this class.[6] Besides their role as marker-passing mechanisms, these nodes or elements also represent concepts in memory. Finally, *value-passing systems* pass continuously-variable quantities among the elements, which in turn perform arithmetic operations on these values. Value-passing systems are equivalent to what Waltz and Pollack call analog spreading activation systems.

The role of spreading activation in AI has been inconsistent at best. Spreading activation was introduced into the AI literature through Quillian's work with semantic memory (see Quillian (1968), for example) but, as Charniak (1983) has observed, this work appears to have been largely ignored by AI researchers through the next decade. Despite this lukewarm reception, the 1970's did produce two noteworthy systems based on spreading activation: ACT, Anderson's general purpose model of human cognition (Anderson, 1976) which has since evolved into ACT* (Anderson, 1983), and NETL, Fahlman's deductive knowledge-base system (Fahlman, 1979).

During the past few years, though, there has been a marked increase in the popularity of spreading activation among AI researchers, especially those building cognitive models of language understanding. There are several reasons why spreading activation schemes are growing in popularity among these researchers; some of

[6]While Fahlman says that markers are passed between processing elements, Waltz and Pollack's definition mentions markers being passed between nodes in a relational network. For the purposes of this dissertation, this distinction will be ignored and the nodes described by Waltz and Pollack's definition are assumed to be the simplest of marker-passing processing elements.

the reasons apply because of the advantages inherent in any kind of parallel processing, while others are relevant because of specific attributes of spreading activation algorithms.

Neuropsychological evidence

The decision to base ATLAST's architecture on a spreading activation mechanism was motivated primarily by recent psycholinguistic studies of the lexical disambiguation process, discussed previously in Chapter 2. These studies led to the conclusion that when an ambiguous word is read or heard in context, all meanings (or senses) of the word are initially primed or recalled, with context then being consulted to select the correct meaning. Spreading activation proves to be an excellent computational mechanism for modeling the lexical access process just described; further, it readily serves as the foundation for a higher-level, modular, parallel-process model of language understanding.

In addition, lower-level neural network models of cognition that rely on analog spreading activation have recently grown in popularity among artificial intelligence and cognitive science researchers because it is generally accepted that, whatever it is that the brain does at its lowest levels, the brain does it in parallel. As Feldman and Ballard (1982) point out, the computational speed of neurons is a few milliseconds, but the action of these neurons must account for complex behaviors that are carried out in a few hundred milliseconds, indicating that these slow, simple neurons must be acting in parallel. A review of the neuropsychological evidence favoring parallelism in the brain is given by Anderson and Hinton (1981).

Shifting the focus to processing

Another advantage of the spreading activation paradigm is that it encourages the researcher to shift the emphasis from knowledge representation to processing mechanisms. Many NLU models concentrate on issues of representation while ignoring issues of processing. Moyne and Kaniklidis (1981, p. 268) offer this assessment of the state of the art in 1981:

> We cannot fail to observe that a rather alarming number of models of comprehension have virtually nothing specific to say about processing mechanisms. Furthermore, what they do have to say, as for example in some vague theorizing about knowledge being stored in the form of labeled graphs, often amounts to little more than the claim that the relevant knowledge is organized (surely an innocuous enough claim) without any precise specification of how it is organized beyond the indication that the units of knowledge are interconnected somehow and that certain broad relations obtain among them.

Adding to the criticism of these same models, Norvig (1983, p. 284) says: "This preoccupation with knowledge structures can sometimes lead to programs with impoverished, redundant, or inconsistent processing mechanisms."

A spreading activation model essentially prevents the researcher from entirely ignoring processing issues, though it is still possible to pay them less attention than they deserve. Thus, spreading activation models may help bring about a change in methodology which is long overdue in artificial intelligence and cognitive science.

Speed of processing

One of the benefits which derives from any form of parallel processing is an increase in the number of computations which can be performed in a given amount of time. Spreading activation provides the ability to explore a number of alternatives (e.g., word senses, syntactic structures, semantic interpretations) at the same time, instead of exploring one at a time, thereby increasing the processing speed of the system. The speed advantage of spreading activation systems over their serial counterparts should be great, so long as there are massively-parallel computer architectures around to support the algorithms. Unfortunately, with few exceptions (e.g., Hillis, 1985) such machines are not yet widely available.

Opening new doors

The tools which a researcher brings to bear on a problem directly influence his or her thinking about the problem. Spreading activation, viewed as one of these tools, enables NLU researchers to propose new, and potentially better, models of language processing. This conclusion is a subjective one, but it is shared by other AI and cognitive science researchers.

One perspective comes from Rieger, Trigg, and Bane (1981, p. 955), who argue the need for parallel computing architectures in AI research. Though the focus of the argument is on actual hardware, the relevance of the argument becomes obvious if one merely substitutes the word "tools" for "computing hardware":

> Beliefs about the nature of [the] computing hardware [which is] available influence a researcher's ability to conceptualize data and process models of intelligence. While a researcher can certainly imagine methods of modeling which require unusual hardware, he may never discover the interesting issues because of his inability to see beyond the first round of ideas. Like it or not, AI is an experimental science which relies heavily on feedback from the implementation level to the conceptual model level.

Closer to the issue at hand are the observations of Pylyshyn (1980, p. 124):

> Now, what is typically overlooked when we [use a computational system as a cognitive model] is the extent to which the class of algorithms that can even be considered is conditioned by the assumptions we make regarding what basic operations are possible, how these may interact, how operations are sequenced, what data structures are possible, and so on. Such assumptions are an intrinsic part of our choice of descriptive formalism.

Finally, Feldman and Ballard (1982, pp. 206-207) offer similar justification for research specifically into analog spreading activation models, but which may be just as relevant to digital spreading activation models:

> The most important reason for a serious concern in cognitive science for [connectionist models] is that they might lead to better science. It is obvious that the choice of technical language that is used for expressing hypotheses has a profound influence on the form in which theories are formulated and experiments undertaken. Artificial intelligence and articulating cognitive sciences have made great progress by employing models based on conventional digital computers as theories of intelligent behavior. But a number of crucial phenomena such as associative memory, priming, perceptual rivalry, and the remarkable recovery ability of animals have not yet yielded to this treatment. . .

Paraphrasing these researchers, it seems that without new tools and the new ways of thinking that go with them, interesting issues go unexplored, and important answers go undiscovered. If for no other reason than this, the different spreading activation techniques are tools which deserve careful consideration.

3.4 Conclusion

The purpose of this chapter has been to introduce the fundamental principles of ATLAST's various components: the relational memory, the syntactic parsing element in the capsulizer, the spreading activation search carried out by the proposer, and the thematic role filling and inference evaluation capabilities of the filter. In other words, this chapter focused on the "how" of ATLAST; Chapter 4 takes an in-depth look at the "why."

Chapter 4

Lexical Inference Processing: Theory

4.1 An unanswered question

The inference processing mechanism implemented in ATLAST is based on a theory of lexical disambiguation called *conditional retention.* The theory of conditional retention was developed in response to questions that are not answered by prevailing psycholinguistic theories of lexical disambiguation. This chapter describes the theory of conditional retention in detail and discusses the arguments for and against this theory.

The active suppression theory of lexical access and disambiguation, introduced in Chapter 2, proposes that when an ambiguous word is read, all meanings are accessed at once and shortly thereafter the meaning which best fits with the existing context is chosen and the alternate meanings are forgotten or suppressed (Seidenberg et al., 1982; Tanenhaus et al., 1979). Although the active suppression theory has been well received by some psycholinguists and has influenced several AI models of language understanding (e.g., Charniak, 1983; Cottrell & Small, 1983; Gigley, 1983; Hirst, 1984; Waltz & Pollack, 1985), the theory is not without its shortcomings.

The active suppression theory suggests that no information about previously recalled meanings is preserved once those word meanings have been suppressed. This raises at least one important and unanswered question: if inappropriate meanings are actively suppressed, how is a reader able to find a correct meaning when later text shows that the initial choice of word meaning is incorrect?[1] Without some means to remember these candidate meanings, the active suppression theory seems to imply that recovery from an incorrect choice of word meaning must involve reprocessing of the text and reactivation of all meanings of the ambiguous word.

4.2 Conditional retention offers an answer

One theory which specifically addresses the problem of finding the correct word sense after previously selecting an incorrect word sense is called *conditional retention* (Granger, Holbrook, & Eiselt, 1984). The conditional retention theory proposes that lexical disambiguation is an automatic process in which all meanings of an ambiguous word are retrieved, the meaning most appropriate to the preceding context is chosen, and the other, less appropriate meanings are temporarily deactivated but retained. In the case where the ambiguous word appears within a short text, the meanings are retained until the end of the text. Should later text contradict the initially chosen meaning, the retained meanings for that word are reconsidered in light of the updated context, and a new meaning is selected without repeating the lexical retrieval process. When there is no further text, a meaning is chosen based on previous context and the other meanings are actively suppressed. The theory of conditional retention does not contradict active suppression; instead, conditional retention merely supplements

[1]This referred to by some as garden path recovery. That designation is not used here because of its associations with syntactic problems; this dissertation is concerned primarily with lexical and pragmatic problems.

active suppression by proposing that those word meanings which are not selected may be deactivated but are not immediately forgotten.

The combined deactivation and retention of the meanings not chosen accomplishes two goals for the language understander. First, it permits the processing at the lexical level to continue to make immediate decisions about the meanings of subsequent words in the context of a single, plausible interpretation of the preceding text instead of multiple interpretations of varying plausibility. If the unchosen meanings were not completely deactivated, there could be resulting confusion in making decisions about new word meanings. Second, it allows the retained meanings to be used by other processes in correcting wrong decisions made by the original process without reprocessing the original input text, at least for a short time. Tracking retained meanings allows the error recovery to be done without maintaining separate copies of all possible interpretations of the text processed so far, thus reducing both storage and processing overhead.

The inadequacy of the active suppression theory by itself is illustrated by the following text:

Text 4: The instructor made the medical students examine the tiny cell so that they would be aware of the consequences of Medicaid fraud.

The reader may find this text to be awkwardly contrived; this is done to highlight phenomena which occur frequently but go unnoticed because of their subtlety. In this example, the ambiguous word "cell" is embedded in a text in which the previous context biases for a "biology" meaning and the later context biases for a "prison" meaning. The active suppression account of this text would have all but the "biology" reading of "cell" suppressed well before the end of the text. However, at the end of the text, the "prison" reading is more appropriate. If the "prison" meaning is no longer active, there is no way to make this interpretation, yet it is obvious that the interpretation is possible. Conditional retention permits this interpretation: after "cell" is encountered, the "biology" meaning is active, but the other meaning is retained until the end of the text. In this case, the final word requires a reinterpretation of the early text and the word meaning as well, so the retained meaning ("prison") supplants the initially selected meaning ("biology") and the reinterpretation is completed. Because no text follows the disambiguating word, the other meanings are then actively suppressed.

The key assumption of conditional retention is that there is a simple binary mechanism for temporarily remembering that a meaning was activated, even if activation has since ceased, and that this memory can be erased when necessary. Combining this two-state memory for a given meaning with two levels of activation for that meaning (i.e., "active" and "inactive") theoretically gives four possible states in which any meaning can exist: (1) the meaning is inactive and there is no memory of its having been recently active; (2) the meaning is inactive but there is memory of its having been recently active; (3) the meaning is active and there is memory of its having been active; and (4) the meaning is active but there is no memory of its having been active.

While it is conceivable that a meaning could be active with no record of recent activation, this notion serves no useful purpose in the conditional retention theory. Thus, there are three states in which word meanings exist:

- **Active:** The meaning is active and there is some memory of its having been active. The meaning shows facilitation (i.e., shorter response times and lower error rates on experimental tasks).
- **Inactive:** The meaning is not active and there is no memory of its having been active. This could be the initial state of the meaning, or could result

from either active suppression or gradual loss of activation. In this state, the meaning shows no facilitation.

- **Retained:** The meaning is not active but there is some memory of its having been active, thereby distinguishing it from an inactive meaning and making it readily available for reconsideration if necessary. The meaning does not show facilitation, thus preventing it from being confused with an active meaning.

A theory of different states of activity in memory is by no means unprecedented. For example, Wagner's (1981) model of learning and automatic memory in animals distinguishes between a state of inactivity (I), a primary state of activity (A1), and a secondary state of activity (A2). Nodes in memory make the transition from I to A1 upon being directly activated by some stimulus, then decay to A2, and finally back to I. If an inactive node is indirectly activated through the spread of activation the node can make the transition from I to A2, but spreading activation cannot push a node from I to A1 or from A2 to A1 (Figure 4.1). Wagner also notes that several other theorists discriminate between two different states of activity in memory, dating at least as far back as the previous century (Morgan, 1894/1977).

The various processes involved in lexical disambiguation can be defined easily in terms of transitions between the three states (Figure 4.2). These processes are:

- **Spreading Activation:** The process of changing inactive or retained meanings to active meanings.
- **Active Suppression:** The process of changing active or retained meanings to inactive meanings.
- **Conditional Retention:** The process of changing active meanings to retained meanings. Note that inactive meanings cannot be directly changed to retained meanings.

Thus, in conditional retention theory, terms such as "retained" or "retention" have very specific meanings with respect to word meanings, lexical disambiguation, and related higher-level memory structures and processes.

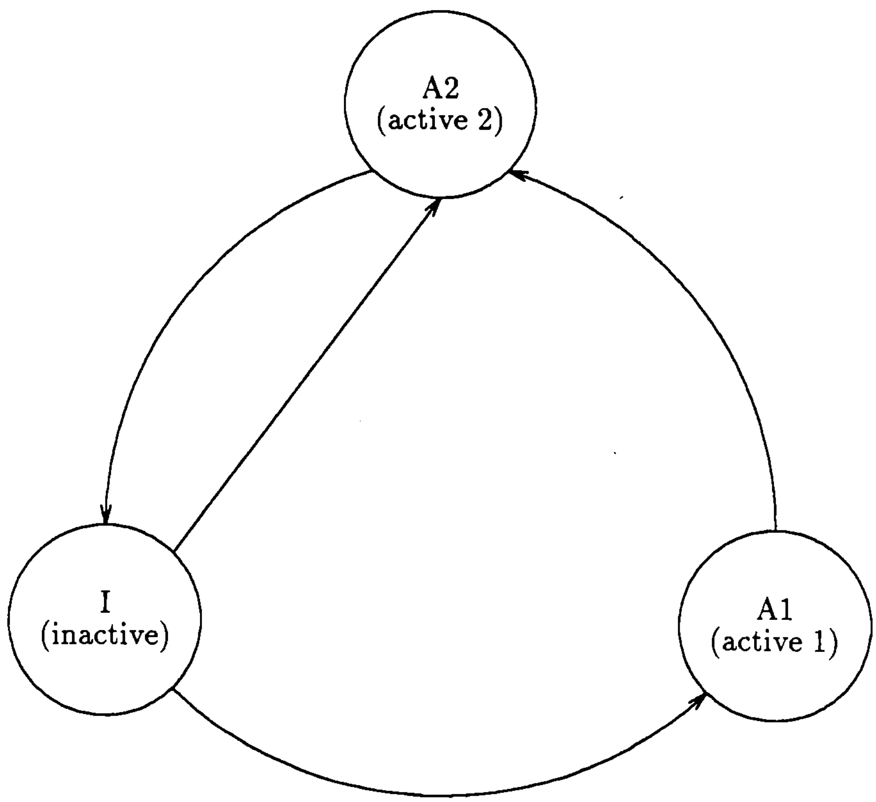

Figure 4.1: Transition network for multiple states of activity in Wagner's (1981) model of learning and automatic memory in animals.

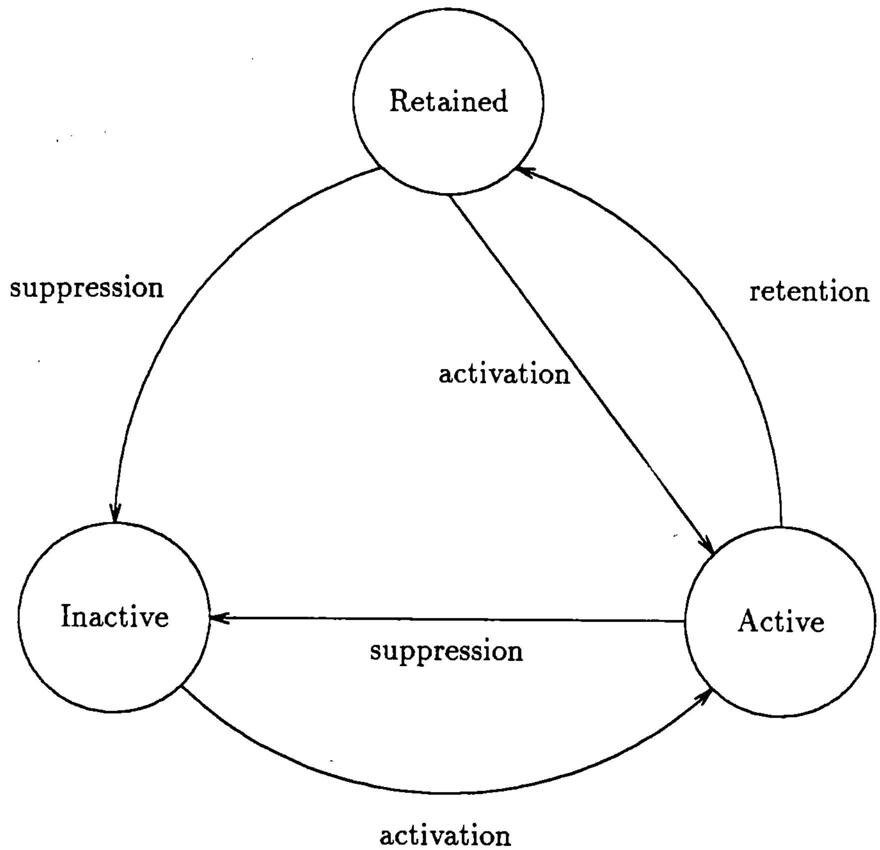

Figure 4.2: Transition network for multiple states of inactivity in the conditional retention theory.

The conditional retention theory extends the active suppression theory by offering an explanation of how early, misleading text can be reinterpreted without conscious reprocessing. Neither active suppression nor any other theory of lexical disambiguation accounts for this phenomenon.

4.3 The controversy

The theory of conditional retention is by no means widely accepted, and the criticisms of conditional retention should be considered when evaluating ATLAST's utility as a cognitive model. One argument against conditional retention is the existence of a large body of experimental evidence which shows that, almost immediately after a meaning of an ambiguous word has been selected, the alternate meanings seem as if they had never been recalled (Onifer & Swinney, 1981; Seidenberg et al., 1982; Swinney, 1979; Tanenhaus et al., 1979). This has been interpreted by some critics as proof that retention does not occur (e.g., G. Hirst, personal communication, March 26, 1986). What seems to be overlooked by the critics, however, is that these experiments simply do not ask the right questions.

4.3.1 Lexical access data: fact and fiction

Critics have argued that the theory of conditional retention runs counter to the data obtained through numerous experiments. The evidence, they say, supports an extreme interpretation of active suppression: that the unchosen meanings of an ambiguous word are immediately deactivated and discarded when one meaning is selected. There is, however, a problem with this interpretation. While they do in fact demonstrate that unselected word meanings are very quickly deactivated, *these experiments do not test whether the meanings have been completely forgotten.*

The experimental technique used in making on-line measurements of the relative activation levels of word meanings during language comprehension is called cross-modal lexical priming (CMLP). In CMLP, the subject is presented aurally with a text containing an ambiguous word. At some controlled interval after hearing the ambiguous word (called the stimulus onset asynchrony or SOA), the subject is presented with a visual task which indirectly measures the degree of activation of the ambiguous word's different meanings. In some experiments the visual task may be a lexical decision task in which the subject must determine whether the visual stimulus is a word or a non-word. Meyer and Schvaneveldt (1971) found that it is easier to decide if a visually-presented target string of letters is a word if a semantically related word is presented prior to the target; this is called priming. Other experiments use a naming task in which the subject must read the target string aloud. In either case, the fundamental assumption is that the extent to which the decision or naming is facilitated corresponds to the degree of activation of the target string's internal representation. The facilitation is measured as the amount of time required for the subject to correctly complete the task (also called the latency). Thus a target word which is semantically related to the ambiguous priming word will be recognized more quickly than a target word which is unrelated to the prime. In turn, an unrelated target word will be easier to recognize than a non-word.

Tanenhaus et al. (1979) used CMLP with a lexical naming task in an investigation of the processing of noun-verb lexical ambiguities in syntactic contexts which biased for either the noun reading or the verb reading (e.g., "I bought the *watch*," or "I will *watch*."). They found that naming latencies were facilitated for both readings immediately after presentation of the ambiguous word (0 msec) regardless of the biasing context, but that only the reading appropriate to the context was facilitated 200 msec later. They concluded that the inappropriate reading was actively suppressed. Seidenberg et al. (1982) used semantically-biasing contexts as well as syntactically-biasing ones and studied noun-noun ambiguities in addition to noun-verb ambiguities. Again the ambiguous word was the final word of the auditory stimulus. With the exception that they found some evidence of selective access in the case of noun-noun ambiguities with strong lexical priming for one interpretation, their results supported the conclusion of Tanenhaus et al. The active suppression hypothesis is further supported by the experiments of Swinney (1979) and Onifer and Swinney (1981), who also employed CMLP but with a lexical decision task. The ambiguous word was embedded *within* a semantically-biasing text and visual targets were presented three syllables after the ambiguous word (750-1000 msec; Swinney, 1979) and at 1500 msec (Onifer & Swinney, 1981). In both cases only the context-appropriate reading was facilitated at the presentation of the target.

Obviously there is substantial cause to conclude that the inappropriate meanings of an ambiguous word in context are deactivated soon after they are initially accessed. This conclusion in no way disagrees with the conditional retention theory. Conditional retention conflicts with active suppression only when the experimental data described above are interpreted as evidence that there is absolutely no memory of the newly-inactive meanings ever having been active. This assumption is unfounded, and the theory of conditional retention specifically states that retained word meanings will exhibit no facilitation. None of the experiments above address the question of how recently deactivated word meanings might differ from those which have been inactive for a much longer duration. This difference might appear as different degrees of sensitivity to re-activation while processing additional input, or some other quality that does not correspond to the relative activation level and is therefore immeasurable by the lexical access experiments performed to date.

As a simple example of how one might easily be led to the wrong conclusion by asking relevant questions but never asking the right question, consider the simple incandescent light bulb. Ostensibly, the bulb is either "on" (emitting light) or "off" (no light). A bulb which has just been turned off does not appear to be different from one which has been off for days. In both cases, measuring radiation in the visible light spectrum (i.e., looking at the bulb) shows that neither bulb is giving off radiation. Similarly, measuring electrical current in the power cord shows no flow of electrons. Two entirely different tests reveal no difference between the bulbs. However, if we run a third test on the bulbs, say by using a thermometer to measure the respective temperatures of the two light bulbs, we find a substantial difference: one is warm but the other is cool.

Of course, word meanings are not light bulbs, nor is conditional retention necessarily related to the dissipation of heat from a light bulb. The point of the light bulb example is that the data from an experiment depends on the measurements made and the tools used to make those measurements. Those choices in turn may depend on the questions being asked. Previous experiments in lexical access were not designed to address questions about conditional retention, so it is premature to dismiss the theory on the basis of those experiments.

4.3.2 The challenge of reprocessing

An apparent challenge to the theory of conditional retention comes from the study of human readers' eye fixations during the reading of misleading texts. While the data do not refute the existence of conditional retention itself, they appear to cast some doubt on one of the fundamental assumptions of the conditional retention theory: the assumption that error recovery is done without reprocessing the input.

The following passage was used in an experiment by Carpenter and Daneman (1981, p. 137):

> The young man turned his back on the
> rock concert stage and looked across the
> resort lake. Tomorrow was the annual one-day
> fishing contest and fishermen would invade
> the place. Some of the best bass
> guitarists in the country would come to this
> spot. The usual routine of the fishing resort
> would be disrupted by the festivities.

Subjects in this experiment were asked to read passages such as the one above while the duration and location of their eye fixations were automatically recorded. In the example above, most readers initially interpreted the word "bass" as a kind of fish because the preceding text is biased toward this interpretation. The interpretation is contradicted, however, by the next word, "guitarists," which forces a reinterpretation of "bass" as a low-frequency musical note.[2]

According to the conditional retention theory, this detection of an incorrect lexical inference and subsequent correction should be performed without reprocessing the passage. Yet Carpenter and Daneman found that most readers' eyes fixated on "bass," then moved forward to and fixated on "guitarists," then *regressed* back to

[2]The experimenters intentionally placed the ambiguous word at the end of a line of text, as with the word "bass" in the passage above. It may be the case that this positioning of the ambiguous word plays a role in eliciting the backtracking effect observed by Carpenter and Daneman, although this is not clear.

"bass," moved forward to "guitarists" again and continued reading the remainder of the passage.

At first, Carpenter and Daneman's conclusion that readers reprocess the text when they encounter contradictions seems to indicate that the assumption of error recovery without reprocessing is incorrect and, consequently, that the conditional retention theory is superfluous. The reprocessing heuristic is just one of several error recovery heuristics proposed by Carpenter and Daneman, however. Another heuristic involves making a larger-than-normal inference encompassing both the inconsistent concept with the preceding text. This is done, they say, if the contradiction is only "mildly semantically inconsistent" and does not involve a syntactic inconsistency (Carpenter & Daneman, 1981, p. 141).[3] This description appears to fit the conflict generated by either Text 3 or Text 4, which are repeated below:

Text 3: Officials at the U.S. Embassy in Moscow have called for a specialist to rid the new building of bugs. Secretaries there have reported seeing cockroaches in the employees' cafeteria.

Text 4: The instructor made the medical students examine the tiny cell so that they would be aware of the consequences of Medicaid fraud.

Intuitively, both texts seem much less likely to cause the degree of confusion observed by Carpenter and Daneman (as reflected by the duration of regressive fixations). These texts are therefore more suited to error recovery by making a more encompassing inference. Error recovery in the conditional retention framework may also be viewed as the making of more encompassing inferences to integrate conflicting concepts; the difference is that Carpenter and Daneman theorize that the more recent text is reinterpreted in such a way as to permit the original misinterpretation to

[3]Still another error recovery heuristic is to continue reading the text with the expectation that later information will resolve the inconsistency. This is similar to the *recency* inference strategy discussed in Chapter 6.

be maintained, while the conditional retention theory allows for previously retained inference possibilities to be re-evaluated without reprocessing the actual text so that the obvious interpretation of the recent text is upheld.

In addition to permitting error recovery without reprocessing, Carpenter and Daneman's theory also demonstrates the need for retention of some sort. Their proposed heuristic of checking the previous text for words that caused processing difficulties, such as ambiguous words, does not specifically address how the reader knows which word or words to reread, but Carpenter and Daneman theorize that difficulties encountered during processing may leave a memory trace which makes finding the ambiguous word much easier. The Carpenter and Daneman model follows the premise that the activation levels of those concepts not selected for use in the interpretation either decay or are actively dampened to a base level, which would preclude the possibility that the memory trace is represented as activation, so this model strongly suggests a retention mechanism which is related to but distinct from the activation mechanism.

4.4 Evidence for conditional retention

4.4.1 Holbrook and Eiselt

A study was designed to test explicitly for conditional retention. This experiment, by Holbrook and Eiselt (in preparation), uses a variable-delay forced-choice task to study the selection and retention of meanings of ambiguous words embedded in texts with variable biasing information before and after the ambiguous word.

Most recent studies of meaning selection have used either lexical decision tasks or lexical naming tasks within a cross-modal lexical priming paradigm, as described earlier in this chapter. The forced-choice paradigm used by Holbrook and Eiselt is an unusual design for a psycholinguistic experiment, although it is frequently used in perception experiments. Lexical decision tasks and lexical naming tasks are good for acquiring data about subjects' response times, which in turn correlate to the degree of facilitation of the individual word meanings. One aspect of the conditional retention theory, however, is that retained meanings may not exhibit any measurable amount of facilitation. This suggests the need for an experimental methodology other than the lexical decision or lexical naming task. The forced-choice task offers the ability to detect indirectly the existence of conditional retention by studying the subjects' decisions instead of their response times.

In this experiment, subjects were asked to read short texts of one to three sentences in length. Experimental texts were two sentences in length, and the others were filler texts. There were two types of experimental texts that are important to the current discussion:[4]

- **Consistent bias surrounds ambiguous word:** The context which precedes the ambiguous word biases toward one of its meanings, but does not preclude the possibility of its other meaning. The second sentence contains information which disambiguates the ambiguous word with information that is consistent with the context of the first sentence, agrees with the original meaning selection, and precludes the other meaning.

  Example: Mary realized that she had examined the wrong bat. She took it back and got one that was aluminum.

- **Conflicting bias surrounds ambiguous word:** The context which precedes the ambiguous word biases toward one of its meanings, but does not preclude the possibility of its other meaning. The second sentence contains information which disambiguates the ambiguous word with information that is consistent

[4]This experiment is discussed in much greater detail by Holbrook and Eiselt (in preparation) and Holbrook (in preparation).

with the context of the first sentence but disagrees with the original meaning selection and requires that the unselected meaning be integrated into the text, precluding the original meaning selection.

Example: Mary realized that she had examined the wrong bat. She took it back and got one that was male.

The texts were presented on a computer monitor a few words at a time, with each group of words replacing the group before it. An information probe was displayed on the monitor at one of two points in the text: either between the ambiguous word and the disambiguating text or after the disambiguating text. The probe consisted of a pair of words, and the subject's task was to decide which of the two words was more related to the text. The choice was indicated by pressing one of two buttons, each corresponding to one of the two probe words. An example of materials presentation is shown below:

Mary realized
that she had examined
the wrong bat. She took
it back and got
CAVE   PITCH
one that was
male.

The first line would appear, centered on the monitor, for 640 msec. The second line would then replace the first line for 640 msec, and so on. When the two capitalized probe words appeared on the screen, the subject would press a designated key on the left side of the computer keyboard if he or she thought the word on the left was more appropriate to the text than the word on the right, or would press a designated key on the right side of the keyboard if the word on the right was thought to be more appropriate. Presentation of additional text did not continue until the subject pressed one of the two keys.

An early probe point and a late probe point were used for each text in a between-subjects design. The early probe point occurred some time after the ambiguous word but before the disambiguating information in the second sentence. The late probe point occurred at the end of the second sentence, after the disambiguating information had been presented. The probe word pairs were rotated between three types of words: a word related to the meaning of the ambiguous word that was correct at the end of the text, a word related to the incorrect meaning at the end of the text, and a word unrelated to either meaning or to the text as a whole. In the example above, the three words used were "PITCH," "CAVE," and "FRUIT." "PITCH" is related to the baseball meaning of "bat," and is more appropriate at the early probe point.[5] "CAVE" is related to the animal meaning of "bat," and is more appropriate to the text at the late probe point. "FRUIT" is unrelated to either meaning of the ambiguous word or the text.

Case I: Consistent bias at the early probe point

In this experiment, the theories of conditional retention and active suppression are set up as opposing theories, so it is useful to compare the two theories' predictions of the outcomes for the different conditions of this experiment. The predictions of the subjects' responses at the early probe point, and the data gathered at this point are summarized in Table 4.1. This table refers to three types of word stimuli. Correct word stimuli are words related to the meaning of the ambiguous word that could be integrated with context at the conclusion of the text. Incorrect word stimuli are

[5] An independent group of informants read the first sentence of each experimental text, which was followed by the two target words that were semantically related to the ambiguous word at the end of the sentence. The informants were asked to choose the word that was more related to the meaning of the sentence, or to indicate that both words were equally related. A sentence was considered to bias toward one of the target words if that word was chosen by the informants more than 80% of the time.

consistent bias surrounds (n=18)	Cell 1		Cell 2		Cell 3	
	correct	incorrect	correct	unrelated	incorrect	unrelated
CR	75	25	100	0	100	0
AS	100	0	100	0	50	50
data	61	39	89	11	94	6
conflicting bias surrounds (n=27)	**Cell 4**		**Cell 5**		**Cell 6**	
	correct	incorrect	correct	unrelated	incorrect	unrelated
CR	25	75	100	0	100	0
AS	0	100	50	50	100	0
data	33	67	70	30	85	15

Table 4.1: Predictions and results at early probe point during forced choice task.

words related to the meaning of the ambiguous word that could not be integrated with context at the conclusion of the text. Unrelated word stimuli are words that were unrelated to either meaning of the ambiguous word at any point in the text. The tables give predicted choices as percentages of the total number of responses.

The first type of text reported in Table 4.1, called "consistent bias surrounds," was designed so that the context which occurs before the ambiguous word is encountered biases towards one meaning of the ambiguous word. Thus, the reader will have enough information from the text on which to base a decision, and will choose the meaning which is more related to the previous context. The context which follows the ambiguous words for these texts agrees with the context that precedes the ambiguous words, so the meaning choice that was made remains correct throughout the text.

Conditional retention (CR) predicts that the unselected meaning of the ambiguous word will be retained throughout the text. Retention of the unselected meaning will cause interference in the forced-choice task between the correct and incorrect probe words (Cell 1): the retained meaning will give the incorrect probe word some non-zero probability of being selected, although the probability is less than that of the correct probe word being selected. The simplest prediction about this interference is that it will be reflected in 25% of the responses (the intermediate point between a prediction of no interference, which would be reflected in 0% of the choices, and complete interference, which would be reflected in 50% of the choices). However, when the correct word is paired with the unrelated word (Cell 2), there is no reason to select the unrelated word over the correct word; the unrelated word was never considered and is not being retained. Therefore, the correct word should always be chosen in this condition. When the incorrect word is paired with an unrelated word (Cell 3), the incorrect word will always be chosen by virtue of its relationship to the retained meaning of the ambiguous word.

Active suppression (AS) makes a different set of predictions. When a meaning for the ambiguous word is selected, the unselected meaning will be actively suppressed. At the point of the forced-choice task, the suppressed meaning should have no effect on the word chosen in the task. Thus, when the choice is between correct and incorrect probe words (Cell 1), active suppression predicts that the correct word will be chosen 100% of the time. There is no tendency to choose the incorrect word because it is associated with the unselected meaning of the ambiguous word, which was previously suppressed and causes no interference. When the choice is between the correct probe word and the unrelated word (Cell 2), the subject again will always choose the correct word. When the incorrect word is paired with the unrelated word (Cell 3), there is no reason to suppose that the incorrect word is chosen with more probability than the unrelated word.

Sign tests were performed on individual cells to determine the significance of the differences between the two stimuli presented to the subjects in each condition. A sign test is a nonparametric statistical test that is useful when the data can be defined as the difference between a pair of qualitative measurements. The sign test simply determines whether two population distributions are identical; in this case, the population distributions are the proportion of one answer to the other in each cell.

In these data, there are some cells that have statistically insignificant results, according to the sign test, which nevertheless seem better predicted by one theory or the other. To confirm such trends in the data, a binomial probability distribution model of the data was used to compare the predictions of each theory to the data. This was done by taking the data in a cell and finding the probability that the prediction made by a theory is matched by the data.

For Cell 1, the sign test of the data did not show a significant difference between the null hypothesis that there is no effect of context and stimulus type and the theories' predictions that there will be some difference. However, the raw data and predictions of each theory seem to favor the conditional retention theory. Using the binomial probability distribution model to compare the two theories to the data, it was determined that the conditional retention theory's prediction for Cell 1 approaches significance, while the active suppression theory's prediction for this cell does not.

In Cells 2 and 3, the sign test found significant differences in the proportions of subjects' responses to the stimuli. The predicted proportion for Cell 2 was the same for both theories, so the results in this cell do not support one theory over another. However, for Cell 3, the predicted proportion for conditional retention was almost exactly matched. The prediction made by active suppression was not supported at all.

Case II: Conflicting bias at the early probe point

The other type of text reported in Table 4.1, called "conflicting bias surrounds," was designed so that the context which occurs before the ambiguous word is encountered biases towards one meaning of the ambiguous word. During the meaning decision process for the ambiguous word, the meaning that fits better with the previous context will be chosen. The context which follows the ambiguous words for these texts disagrees with the context that precedes the ambiguous words, so the meaning that is contextually appropriate at the early probe point will be inappropriate at the end of the text. Correspondingly, the probe word that is correct at the end of the text in in "consistent bias" texts is incorrect at the end of "conflicting bias" texts, and the probe word that is incorrect in the former case is correct in the latter.

Because the forced-choice task is presented before the contradiction is encountered, the two theories' predictions of the subjects' choices are exactly the same as those in Case I above, but the designations of "correct" and "incorrect" are reversed. Thus, the predictions in Cell 4 are the reverse of Cell 1, the predictions of Cell 5 are the same as those of Cell 3, and the predictions in Cell 6 are the same as in Cell 2.

In Cell 4, conditional retention and active suppression both predict a significant effect of context type and target type. The null hypothesis predicts that no such effect will obtain. The sign test performed did not yield results that make it possible to reject the null hypothesis. However, a binomial probability model used for analysis indicates that the probability of the conditional retention theory matching the observed data because of effect rather than chance is far greater than for the active

consistent bias surrounds (n=18)	Cell 7		Cell 8		Cell 9	
	correct	incorrect	correct	unrelated	incorrect	unrelated
CR	100	0	100	0	50	50
AS	100	0	100	0	50	50
data	100	0	94	6	61	39
conflicting bias surrounds (n=27)	Cell 10		Cell 11		Cell 12	
	correct	incorrect	correct	unrelated	incorrect	unrelated
CR	100	0	100	0	100	0
AS	0	100	50	50	100	0
data	59	41	89	11	78	22

Table 4.2: Predictions and results at late probe point during forced choice task.

suppression theory. In other words, although the sign test does not show a significant effect, the conditional retention theory provides a far better match to the data than does the active suppression theory.

In Cells 5 and 6, the sign test found a significant difference between the proportions of the subjects' responses to the different stimuli. Conditional retention predicted this difference in Cell 5, but active suppression did not. In Cell 6, both theories predicted this difference.

Case III: Consistent bias at the late probe point

The predictions for the subjects' responses at the end of the experimental texts, and the subjects' responses at that point are shown in Table 4.2. In the case of the "consistent bias surrounds" texts, the ambiguous word is encountered at the end of the first sentence, but the forced-choice task is not presented until the end of the second sentence, which reinforces the contextual bias of the first sentence.

One question that remains unanswered by the conditional retention theory is, "For how long are unselected meanings retained?" As described in the following chapter, ATLAST places a limit on the number of times that a retained inference path, and the word sense contained within that path, can be re-evaluated. ATLAST also suppresses all retained paths at the end of the text it is processing. However, recent experimental data suggests that the duration of retention may be affected by confirming or disconfirming evidence which follows the ambiguous word (Holbrook, in preparation). The predictions for conditional retention in Table 4.2 reflects the latter idea about retention duration. Thus, for "consistent bias surrounds" texts, conditional retention predicts that retained meanings will be suppressed by the late probe point because of the confirming evidence of the second sentence. Active suppression predicts that unselected meanings will be suppressed at the end of the first sentence. Consequently, the conditional retention and active suppression theories predict the same results for all three experimental conditions in the case of "consistent bias surrounds" texts, as explained below.

Given a choice between the correct probe word and the incorrect probe word (Cell 7), both theories predict that the subject will always choose the correct word. The active suppression theory predicts this result because the unselected meaning of the ambiguous word, which is related to the incorrect probe word, will have been suppressed shortly after the ambiguity was encountered. The conditional retention theory proposes a different course of events with the same result. The unselected meaning is retained for a period of time during the reading of the second sentence. The second sentence reaffirms the subject's decision about the meaning of the ambiguous word and forces suppression of the retained meaning. Both theories predict that the subject will select the correct word over the unrelated word (Cell 8). Given a choice between the incorrect word and the unrelated word (Cell 9), both theories predict that there is an equal probability of choosing either word.

The data supported the predictions in Cells 7 and 8 that there would be significant differences in the subjects' responses. The data also seem to support the prediction in Cell 9 that there would be no significant difference in this condition, but the analysis of the data did not reject the possibility that a significant difference does exist.

Case IV: Conflicting bias at the late probe point

In the case of "conflicting bias surrounds" texts with the forced-choice task presented at the end of the text (Table 4.2), the differences between the two theories become apparent once more. Here, as in Case II above, the second sentence contradicts the subject's meaning selection for the ambiguous word at the end of the first sentence. In order to interpret the text correctly, the subject must revise his or her original meaning selection for the ambiguous word. Unlike Case II, however, the forced-choice task is presented *after* the subject has read the contradictory text. This contradiction discourages suppression of the unselected but retained meanings, which in turn allows the subject to revise the meaning selection. The effects of this revision should be apparent in the data if the conditional retention theory is correct, but the revision should not be reflected in the data if the active suppression theory is correct.

When presented with a choice between the correct probe word and the incorrect probe word (Cell 10), the conditional retention theory predicts that the subject will choose the correct word at the late probe point. The active suppression theory, on the other hand, predicts that the subject will choose the incorrect word, because this word is related to the only meaning of the ambiguous word that is available to the subject at this point: the meaning that was appropriate at the end of the first sentence but is now inappropriate.

Conditional retention predicts that the subject will choose the correct word over an unrelated word (Cell 11). Active suppression, however, predicts that there will be an equal probability of choosing either the correct word or the unrelated word. This should occur, if the active suppression theory is correct, because the probe word that is correct at the end of the second sentence is semantically related to the meaning of the ambiguous word that was *not* selected at the end of the first sentence and has now been suppressed. From an active suppression standpoint, a word that is related to a suppressed meaning is no more likely to be chosen than a word which is not related to the text.

Finally, both theories predict that the subject will select the incorrect probe word over the unrelated word (Cell 12). The conditional retention theory makes this prediction because the incorrect word is semantically related to the meaning of the ambiguous word that was originally selected but is now retained. The active suppression theory makes this prediction because the incorrect word is semantically related to the ambiguous word meaning that was first selected and is still active, all other meanings having been suppressed.

Although the raw data in Cell 10 lean toward the conditional retention theory, the data are not sufficient to reject the hypothesis that there is no significant difference between the choice of the correct and incorrect probe words. Using a binomial probability model, neither theory provided a good fit with the data.

Cells 11 and 12 provided a better fit with the existing theories. For Cell 11, a significant difference between the word choices obtained. Conditional retention predicted that there would be such a significant difference, but active suppression predicted that there would be no significant difference. For Cell 12, a significant difference between the word choices also obtained. This was predicted by both theories.

Discussion of the results

The two theories predicted different results in six of the twelve different test conditions shown in Tables 4.1 and 4.2: Cells 1, 3, 4, 5, 10, and 11. Analysis of the data showed that the conditional retention theory predicted the results better than the active suppression theory did in five of the six cells. The analysis of the data in Cell 10 provided no support for either theory. It is not surprising, however, that this was so, because the forced-choice task in this condition should have been the most difficult for the subjects. This was the only condition in which the forced-choice task came after the contradictory text and both probe words were related to meanings of the ambiguous word. Consequently, the processing load here would be greater than in any other condition.

Overall, the weight of the evidence from this experiment clearly supports the conditional retention theory. The evidence from another experiment, which is discussed next, also lends support to conditional retention, although it suggests a mechanism which differs from the one proposed in this dissertation.

4.4.2 Burgess and Simpson

A different view of retention in language understanding is offered by Burgess and Simpson (1988). They studied the contribution of each hemisphere of the brain to lexical ambiguity resolution. While it is well accepted that the left hemisphere has a major role in language processing, the role of the right hemisphere is unclear. Burgess and Simpson found that the time course of activation of the meanings of an ambiguous word differed profoundly between the two hemispheres.

Subjects in Burgess and Simpson's experiment were presented with lexical decision tasks using a divided visual field methodology. Subjects were visually primed with an ambiguous word displayed on a computer terminal for 35 msec. The priming word was masked for either 0 or 715 msec, after which the target word was displayed on the terminal. The target was visible for 185 msec, then was masked for 50 msec, and then the screen went blank. The targets were related to the prime through the dominant or subordinate meaning of the prime. The twist to this experiment is that the target word was randomly presented either 2 degrees to the left or right of the point where the prime appeared. The duration of the target word's appearance on the screen is too brief to permit the eyes to move to the target, so a target presented on the left side of the screen appears only in the left visual field and is projected to the right hemisphere. Conversely, a target on the right side of the screen appears only in the right visual field and is projected to the left hemisphere.

Burgess and Simpson found that the time course of activation for the different meanings in the left hemisphere agreed with the results of another study in which the targets were presented to both visual fields simultaneously (Simpson & Burgess, 1985): the dominant or more frequently used meaning was facilitated at both 35 and

750 msec, while the subordinate or less frequently used meaning showed less facilitation at 35 msec and no facilitation at 750 msec. In fact, at 750 msec the subjects' responses to related subordinate targets were *slower* than their responses to unrelated trials, leading to the conclusion that the subordinate meaning was inhibited or actively suppressed by that time. In the right hemisphere, on the other hand, facilitation for the dominant meaning *decreased* between 35 and 750 msec while facilitation for the subordinate meaning *increased* significantly over that same time period.

These results, say Burgess and Simpson, suggest a mechanism in which the left hemisphere calls upon the right hemisphere to access memory information when necessary. But when would such a mechanism be needed? Burgess and Simpson speculate that when an ambiguous word is embedded in a sentence but the choice of meaning is not constrained by the preceding context, it would be costly for the language understander to allow the subordinate meaning become inactive either through decay or suppression. For example, consider the following sentence (Burgess & Simpson, 1988, p. 419):

> The man stood by the bank for the better part of an hour before catching a fish worth taking home.

The text following the ambiguous word "bank" supports the choice of the subordinate meaning (i.e., land at the edge of a body of water) instead of the dominant meaning (i.e., a place where money is kept). If the subordinate meaning is inactive when the word "fish" is read, that meaning must be reactivated for processing to continue successfully. If, however, that meaning maintains activation in the right hemisphere, the understander could access the correct meaning at the appropriate time without incurring the cost of reactivating the meaning. In short, Burgess and Simpson propose that "the role of the right hemisphere is to provide the less frequent meaning when needed by context after inhibitory processes in the left hemisphere have caused activation for the subordinate meaning to decline" (p. 421).

ATLAST does not specifically account for potential processing differences between hemispheres of the brain. Nevertheless, ATLAST's premise that retention of a word meaning is distinct from its activation is certainly compatible with Burgess and Simpson's theory that activation and retention are fundamentally the same process carried out by different hemispheres. Their finding that the time course of activation when materials are presented to the left hemisphere looks very much like the time course obtained when materials are presented simultaneously to both hemispheres indicates that the right hemisphere's behavior is obscured by that of the left hemisphere; this would explain why retention has gone undetected by lexical access experiments in which the targets are presented to both visual fields simultaneously.

4.5 Conclusion

The theory of conditional retention offers an explanation for psycholinguistic phenomena which most theories of lexical disambiguation fail to address: recovery from erroneous lexical inferences. The one other theory which addresses this issue, that of Carpenter and Daneman (1981), relies on a retention-like mechanism to recover from mistakes while using reprocessing and concedes a non-reprocessing heuristic which also shares some features with conditional retention. As we have seen, the experiments often cited in support of the active suppression theory, and which some might view as arguments against conditional retention, do not test for conditional retention. Neither do the actual results of those experiments contradict any assumptions of the conditional retention theory. Furthermore, the results of two other experiments lend support to the theory of conditional retention. These facts lead to the conclusion that the theory of conditional retention is currently the best available explanation of lexical ambiguity resolution and error recovery in human language understanding.

Chapter 5

Lexical Inference Processing: Implementation

5.1 Lexical disambiguation and recovery

This chapter describes at length the operation of ATLAST as it processes this shortened version of Text 3:

Text 5: The embassy searched for bugs.
The secretaries had seen roaches.

Text 5 is similar to those used in the experiment to test the conditional retention theory (Holbrook & Eiselt, in preparation), which was described in the previous chapter. The texts in this experiment were designed to establish a context that strongly biases for one meaning of a target ambiguous word and then contradicts that meaning, thus forcing the understander to supplant its original choice of meaning with a new one. In the case of Text 5, the context invoked by the reading of "embassy" forces the understander, ATLAST, to choose the "hidden microphones" meaning of the word "bugs." After reading the word "roaches" at the end of the second sentence, however, ATLAST recognizes that its initial choice of meaning is incorrect and that "insects" is the correct interpretation of "bugs."

Text 5 is the subject of considerable discussion throughout this dissertation (it appears again in Chapter 7), as it provides a good illustration of ATLAST's capabilities. It is not, however, the only text on which ATLAST has been tested. Additional examples are shown at the end of this chapter and in Appendix A. As of this writing, ATLAST has been tested successfully on more than sixty texts of one to four sentences in length, based on six different scenarios. What follows is actual run-time output generated by ATLAST during the processing of Text 5. For the sake of brevity, much of the output has been deleted, leaving only the more interesting parts. A diagram of the corresponding memory structure for Text 5 is shown in Figure 5.1.

5.2 ATLAST is fooled

5.2.1 Metrics and parameters

ATLAST is designed to serve as a framework for testing theories of lexical disambiguation and error recovery, not as a robust language processing system. As such, ATLAST has a number of parameters that can be adjusted to manipulate its processing behavior. ATLAST presents a summary of its parameter settings before beginning work on the input text.

```
Input text is:

    The embassy searched for bugs.
    The secretaries had seen roaches.

Ordering of inference evaluation metrics in force:

    MORE-ACTIVATION-METRIC
    SHORTER-PATH-METRIC
```

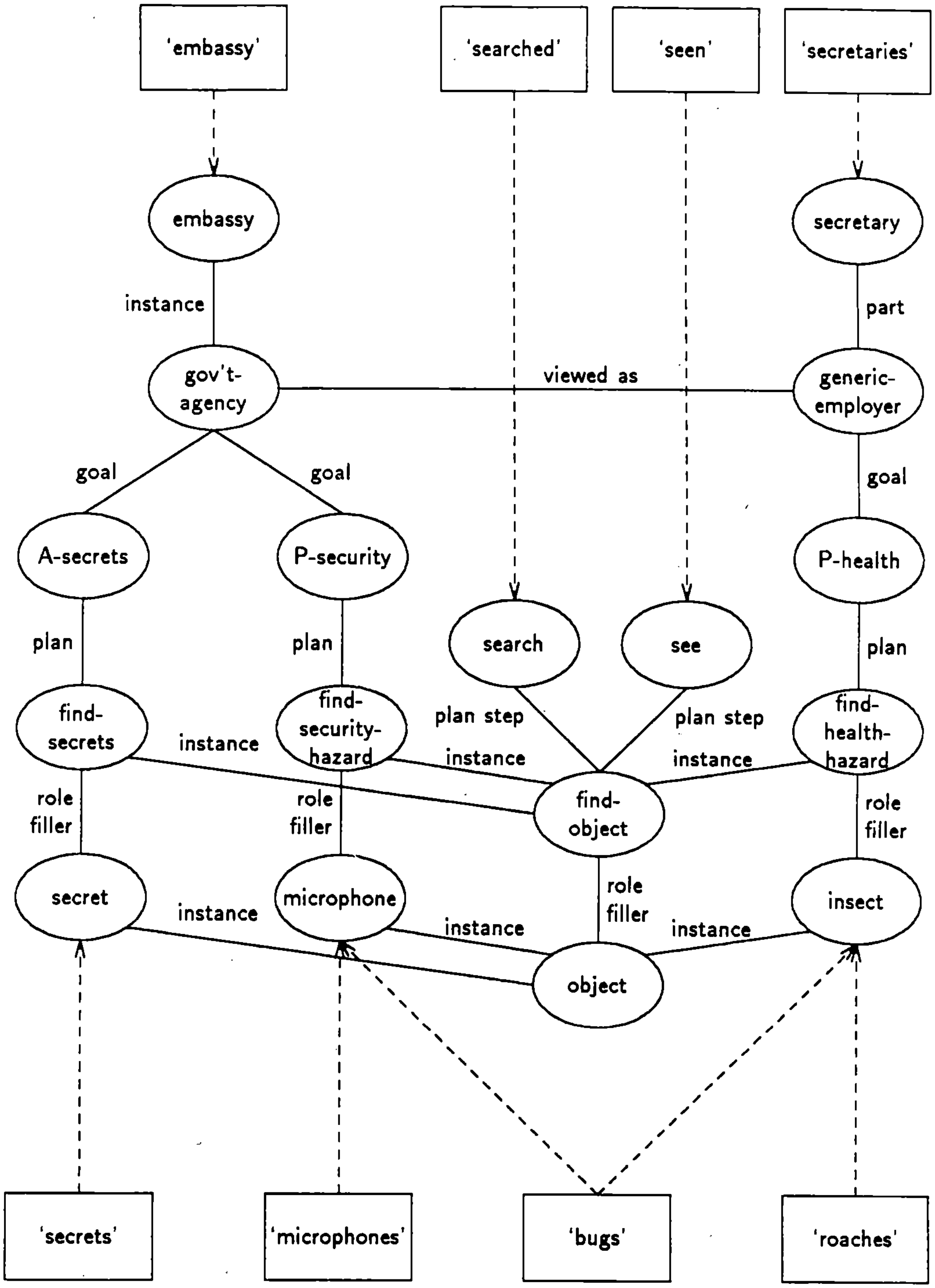

Figure 5.1: Memory network for Text 5.

```
MORE-REINFORCEMENT-METRIC
MORE-SPECIFIC-METRIC
NO-DECISION-METRIC
```

The inference evaluation metrics are used by ATLAST's filter to compare two competing inference paths and select the better one, as described in Chapter 3. The order in which the metrics are applied directly affects ATLAST's interpretation of a given text; the ordering is determined by the user. The ordering of the first four metrics listed above is the only ordering of those metrics that results in acceptable interpretations for the texts on which ATLAST has been tested, but this is not necessarily the one true ordering of evaluation metrics. As ATLAST is tested on more examples, the need for additional metrics or changes to the existing ones will most likely become apparent. As these new or revised metrics are added, different orderings of the metrics may also be required.

The last metric in the list, the *no-decision metric*, is invoked only when the previously applied metrics fail to select one of the competing paths. Therefore, the no-decision metric, which selects neither competing path but retains them both for later re-evaluation, is always the last metric in the ordering.

```
Maximum distance of marker-passing: 3
Distance to pass markers per cycle: 3
```

The two parameters above pertain to ATLAST's proposer. The first parameter indicates that the spread of activation via marker-passing will be limited to a distance of three nodes away from the point of origin. If this limit were decreased, ATLAST would be unable to correctly interpret Text 5 because one of the inference paths essential to understanding this text consists of six links. On the other hand, if this limit were increased, ATLAST would still find the correct interpretation but at the expense of doing a significantly greater amount of work.

The second parameter indicates that markers will be passed a distance of three nodes away from the origin each time that the proposer is invoked (i.e., whenever an open class word is processed). In other words, the values of the two parameters taken together mean that all marker-passing for a given open class word will be done before the next word is processed. While ATLAST works very well when the marker-passing for an input word is performed over multiple cycles while other words are being processed, eye-movement studies have resulted in the conclusion that all the processing for an input word is completed before the next word is read (Carpenter & Daneman, 1981), so ATLAST is working under that constraint.

```
Are rejected paths being retained?: t
Max. no. of unsuccessful evaluations: 5
Min. no. of shared nodes to force re-evaluation: 4
```

This set of parameters controls the retention process that gives ATLAST its error correction ability. The first of these says that inference paths which are evaluated, but not incorporated into ATLAST's interpretation, are to be retained. The second parameter sets a limit on the retention of inference paths. In this case, a retained path may be re-evaluated no more than five consecutive times without being activated. If the path is re-evaluated a sixth time with no success, it is suppressed.

The third parameter determines how retained paths will be chosen for re-evaluation. When the proposer suggests a path to the filter for evaluation, the filter examines the set of retained paths to find those paths that are sufficiently related to the path suggested by the proposer. These related, retained paths then will also be evaluated by the filter. For this example, a retained path must share a subpath of at least four nodes with the path suggested by the proposer to qualify for re-evaluation. (If either path has fewer nodes than the limit, the shorter path must be entirely contained within the longer path for re-evaluation to occur.)

```
Are function words being allocated processing cycles?: nil
Is inference processing forced to complete at periods?: t
```

The last two parameters control how the processing cycles of the proposer and the filter are allocated. The value of the first parameter above means that marker-passing and path evaluation are not performed when function, or closed class, words are processed. The value of the second parameter means that all processing of a sentence is forced to completion at the end of the sentence. In other words, any "loose ends" are resolved as much as possible.

5.2.2 ATLAST begins work

After ATLAST displays the constraints under which it is working, it begins processing the input text. ATLAST attaches a `*START*` symbol to the beginning of a text. Processing this symbol initializes ATLAST.

```
Processing begins

Capsulizer:
  Retrieving lexical entry: *START*
  No nodes will be activated from lexical entry
  Sending capsule: (start)

Filter:

  Received capsule: (start)
  Begin processing of event0
```

ATLAST then reads the first word of the input. No nodes are activated, no markers are passed, and there are no paths to evaluate. The capsulizer's intra-phrasal syntactic analysis suggests that a noun phrase is coming.

```
Capsulizer:
  Retrieving lexical entry: The
```

```
No nodes will be activated from lexical entry
Begin sentence
Begin noun phrase
```

ATLAST now processes **embassy**, the first open class word. The corresponding node in memory is activated and the proposer begins passing markers from there. Because no other open class words have been processed, the proposer cannot find any potential inference paths at this time.

```
Capsulizer:
  Retrieving lexical entry: embassy

Proposer:
  Initializing EMBASSY
    Passing marker0 from EMBASSY to GOVT-AGENCY as marker1
    Passing marker1 from GOVT-AGENCY to A-SECRETS as marker2
    Passing marker1 from GOVT-AGENCY to P-SECURITY as marker3
    Passing marker1 from GOVT-AGENCY to GENERIC-EMPLOYER as marker4
    Passing marker2 from A-SECRETS to FIND-SECRETS as marker5
      marker5 too old to be passed further
    Passing marker3 from P-SECURITY to FIND-SECURITY-HAZARD
        as marker6
      marker6 too old to be passed further
    Passing marker4 from GENERIC-EMPLOYER to P-HEALTH as marker7
      marker7 too old to be passed further
    Passing marker4 from GENERIC-EMPLOYER to SECRETARY as marker8
      marker8 too old to be passed further
```

5.2.3 Proposing possible inference paths

The next word, **searched**, is a verb. This tells the capsulizer that it has reached the end of the noun phrase it was parsing. The capsulizer informs the filter that it has parsed a noun phrase and that the head noun activated the node labeled EMBASSY. The capuslizer now expects a verb phrase.

The proposer passes markers outward from the node SEARCH and finds intersections with the markers passed previously from EMBASSY. These intersections represent paths to be evaluated by the filter.

```
Capsulizer:
  Retrieving lexical entry: searched
  Sending capsule: (nphrase (EMBASSY))
  End noun phrase
  Begin verb phrase

Proposer:
  Initializing SEARCH
    Passing marker9 from SEARCH to FIND-OBJECT as marker10
    Passing marker10 from FIND-OBJECT to OBJECT as marker11
    Passing marker10 from FIND-OBJECT to SEE as marker12
    Passing marker10 from FIND-OBJECT to FIND-SECURITY-HAZARD
        as marker13
      Proposing path from SEARCH to EMBASSY
    Passing marker10 from FIND-OBJECT to FIND-HEALTH-HAZARD
        as marker14
    Passing marker10 from FIND-OBJECT to FIND-SECRETS as marker15
      Proposing path from SEARCH to EMBASSY
    Passing marker11 from OBJECT to MICROPHONE as marker16
      marker16 too old to be passed further
    Passing marker11 from OBJECT to INSECT as marker17
      marker17 too old to be passed further
    Passing marker11 from OBJECT to SECRET as marker18
      marker18 too old to be passed further
    Passing marker11 from OBJECT to LAMP as marker19
      marker19 too old to be passed further
    Passing marker11 from OBJECT to WEAPON as marker20
      marker20 too old to be passed further
    Passing marker11 from OBJECT to POSSESS-OBJECT as marker21
      marker21 too old to be passed further
    Passing marker13 from FIND-SECURITY-HAZARD to P-SECURITY
        as marker22
      Proposing path from SEARCH to EMBASSY
      marker22 too old to be passed further
    Passing marker13 from FIND-SECURITY-HAZARD to MICROPHONE
        as marker23
      marker23 too old to be passed further
    Passing marker14 from FIND-HEALTH-HAZARD to P-HEALTH as marker24
      Proposing path from SEARCH to EMBASSY
```

```
    marker24 too old to be passed further
  Passing marker14 from FIND-HEALTH-HAZARD to INSECT as marker25
    marker25 too old to be passed further
  Passing marker15 from FIND-SECRETS to A-SECRETS as marker26
    Proposing path from SEARCH to EMBASSY
    marker26 too old to be passed further
  Passing marker15 from FIND-SECRETS to SECRET as marker27
    marker27 too old to be passed further
```

The proposer is a relatively boring process. From this point on, everything that the proposer does will look just like what it has done so far, except that different nodes and different markers will be involved, so the remainder of the proposer's output has been omitted.

5.2.4 Evaluating proposed paths

The filter expects that the first noun phrase ATLAST encounters will probably represent the actor in the sentence being processed, so the filter assigns the node labeled EMBASSY to the actor role. The filter also evaluates the paths that were found by the proposer.

```
Filter:
  Received capsule: (nphrase (EMBASSY))
    Filling actor slot in event0 with EMBASSY
```

The first inference path that the filter evaluates is given the name path0. Since no other paths are currently active, path0 has no competitors. Thus, path0 is ATLAST's first addition to the set of active paths.

```
  New path discovered: path0
    Path from SEARCH to EMBASSY
      SEARCH is a plan step of FIND-OBJECT
      FIND-OBJECT has the instance FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is a plan of P-SECURITY
```

```
P-SECURITY is a goal of GOVT-AGENCY
GOVT-AGENCY has the instance EMBASSY
Activating path0
```

The next path, path1, has the same endpoints as path0. Therefore, it competes with the active path0 as a possible explanation of the relationship between the nodes EMBASSY and SEARCH. The filter compares the two paths but its metrics do not prefer one over the other. The no-decision metric is invoked, leaving both paths de-activated but retained. Again, there are no active paths. This is done with the expectation that evaluating other paths during this cycle will alter the context provided by the set of active paths such that a later comparison between path0 and path1 will result in a decision.

```
New path discovered: path1
  Path from SEARCH to EMBASSY
    SEARCH is a plan step of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECRETS
    FIND-SECRETS is a plan of A-SECRETS
    A-SECRETS is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
  No-decision metric -- path0 and path1 are retained
    De-activating path0
    De-activating path1
```

The filter now evaluates path2, which also joins SEARCH and EMBASSY. Although path0 and path1 share the same endpoints as path2, they are not currently active and therefore do not compete with path2. Path 2 is activated, but only briefly as the proposer has rediscovered path1. Now path1 is evaluated against path2, and the filter finds that path1 is more specific than path2. That is, path2 requires that the embassy be viewed abstractly as a generic employer for its explanation of the text to be plausible while path1 requires no such abstraction. Path1 is now added to ATLAST's active interpretation and path2 is de-activated and retained.

```
New path discovered: path2
  Path from SEARCH to EMBASSY
    SEARCH is a plan step of FIND-OBJECT
    FIND-OBJECT has the instance FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is a plan of P-HEALTH
    P-HEALTH is a goal of GENERIC-EMPLOYER
    GENERIC-EMPLOYER can be viewed as GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
    Activating path2
```

However, in the process of re-evaluating path1, the filter is also reminded that an earlier comparison between path1 and path0 resulted in a tie or split decision. Attempting to verify that path1 should be active now, the filter compares it to path0 and again finds that it cannot make a decision between the two.

```
Old path rediscovered: path1
Also reconsidering (path0) due to tie with path1
  More-specific metric -- path1 more specific than path2
    De-activating path2
    Activating path1
  No-decision metric -- path1 and path0 are retained
    De-activating path1
    De-activating path0
```

ATLAST then moves on to the next word, the preposition `for`, which has no semantic representation in ATLAST's memory. As a result, no marker-passing is initiated and no new paths are proposed. This function word does tell the capsulizer that it is now processing a modifying phrase, so the capsulizer informs the filter that it has seen a verb, and that the node labeled `SEARCH` was activated by that verb. The filter then fills the action role with a pointer to that node.

```
Capsulizer:
  Retrieving lexical entry: for
  No nodes will be activated from lexical entry
  Sending capsule: (vphrase (SEARCH))
  Begin prepositional phrase
```

```
Filter:
  Received capsule: (vphrase (SEARCH))
    Filling action slot in event0 with SEARCH
```

5.2.5 Encounter with an ambiguous word

Now, ATLAST reads the word **bugs**. The capsulizer recognizes two different meanings for the word, so the proposer begins searching for possible inference paths from the nodes representing both meanings.

```
Capsulizer:
  Retrieving lexical entry: bugs
  Ambiguous word senses noted: (INSECT MICROPHONE)
```

The filter then begins to evaluate the paths found by the proposer. The first path, path3, has no competing path in ATLAST's interpretation so it is activated.

```
New path discovered: path3
  Path from INSECT to SEARCH
    INSECT is an instance of OBJECT
    OBJECT is a role-filler of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
    Activating path3
```

The next path evaluated, path4, competes with path3. The filter's evaluation metrics cannot determine which path offers the better explanation of the input, so path3 and path4 are both de-activated and retained.

```
New path discovered: path4
  Path from INSECT to SEARCH
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  No-decision metric -- path3 and path4 are retained
    De-activating path3
    De-activating path4
```

The filter now evaluates two paths which connect MICROPHONE to SEARCH. There is no active path competing with path5 so it is added to the interpretation. Path5 now provides competition for the next path to be evaluated, path6. Again the comparison ends in a split decision.

```
New path discovered: path5
  Path from MICROPHONE to SEARCH
    MICROPHONE is an instance of OBJECT
    OBJECT is a role-filler of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
    Activating path5

New path discovered: path6
  Path from MICROPHONE to SEARCH
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  No-decision metric -- path5 and path6 are retained
    De-activating path5
    De-activating path6
```

The filter finds no active path to compete with the next new path, path7, and adds it to the interpretation of the text. The previously retained path0 shares at least four nodes with path7 so it is re-evaluated. Finding no active competing path, the filter activates path0. As before, the filter re-evaluates path1 because of its previous tie with path0 and is still unable to choose one. Path0 and path1 are again de-activated and retained and path7 is the only active path.

```
New path discovered: path7
  Path from MICROPHONE to EMBASSY
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
Also reconsidering (path0) due to shared nodes with path7
Also reconsidering (path1) due to tie with path0
    Activating path7
    Activating path0
```

```
No-decision metric -- path0 and path1 are retained
  De-activating path0
  De-activating path1
```

The next new path, path8, connects the nodes INSECT and EMBASSY, but it competes with path7, which connects MICROPHONE and EMBASSY, because the filter knows that the differing endpoints were activated by the same word, although it does not know what that word was. Thus, the filter regards path7 and path8 as different explanations for the same two words from the input text and evaluates them accordingly. The filter determines that the relationship described by path7 is more specific than that described by path8, so path8 is de-activated and retained. Also, path2 is found to share the requisite number of nodes is path7 and is re-evaluated. Because it has no active competing paths, path2 is activated.

```
New path discovered: path8
  Path from INSECT to EMBASSY
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is a plan of P-HEALTH
    P-HEALTH is a goal of GENERIC-EMPLOYER
    GENERIC-EMPLOYER can be viewed as GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
Also reconsidering (path2) due to shared nodes with path8
  More-specific metric -- path7 more specific than path8
    De-activating path8
    Activating path2
```

At this point, ATLAST's interpretation is inconsistent in that path7 views the embassy as a government agency with a goal of preserving its security but path2 sees the embassy as an employer with a goal of preserving the health of its employees. This inconsistency will be resolved as more retained paths are re-evaluated.

The filter evaluates path9, connecting INSECT to SEARCH, and, finding no active competing path, proceeds to add it to the active interpretation. In evaluating path9, the filter finds that path6 is sufficiently similar to path9 and re-evaluates that path.

Path6 also competes with path9, and the filter determines that path6 is a better choice than path9 because path6 is shorter than path9.

Path5 is re-evaluated because of its earlier tie with path6. Path6 shares more nodes with the other active paths than does path5, so path5 is de-activated and retained. In other words, the combination of path6, path7, and path2 provides a more parsimonious interpretation of the text than path5, path7, and path2 (cf. Granger, 1980b). The active paths are now path2, path6, and path7, but the interpretation is still inconsistent because of the conflicting explanations of the embassy and its goals represented by paths 2 and 7.

```
New path discovered: path9
  Path from INSECT to SEARCH
    INSECT is an instance of OBJECT
    OBJECT has the instance MICROPHONE
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
Also reconsidering (path6) due to shared nodes with path9
Also reconsidering (path5) due to tie with path6
    Activating path9
  Shorter-path metric -- path6 shorter than path9
    De-activating path9
    Activating path6
  More-reinforcement metric -- path6 has more shared nodes
    than path5
    De-activating path5
```

5.2.6 An *is-a intersection*

Path9 is notable also because it contains what others have called an *is-a intersection* (Charniak, 1983) or *is-a plateau* (Hendler, 1986). An is-a intersection exists when two nodes are joined to a third node by is-a links, indicating that the first two nodes represent different instantiations of the concept denoted by the third node.

For example, an is-a intersection would occur in a network which represented that a canary and a robin are two different instantiations of a bird. ATLAST's is-a links are pairs of **has-instance** and **instance-of** links. The is-a intersection in path9 occurs at the node OBJECT. Such intersections may be uninformative, especially if they occur near the top of an is-a hierarchy where nodes may be linked to many other nodes. Marker-passing systems which allow markers to travel in only one direction along is-a links, such as DMAP (Riesbeck & Martin, 1986), are not bothered by paths with is-a intersections because they never find them. Hirst's ABSITY (1988b) protects itself from being swamped by an overabundance of these paths through the use of an *anti-promiscuity rule* which prevents markers from propagating from nodes with more than some small number of links; this allows is-a intersections to occur only in the lower portions of an is-a hierarchy where they will be more specific and more informative. Hendler (1986) recognizes that these intersections may be useful in language understanding, but he says they serve no useful purpose in problem solving, and his SCRAPS problem solving system rejects any path containing an is-a intersection. At this time, ATLAST gives no special attention to paths with is-a intersections because the evaluation metrics so far have eliminated such paths from the interpretation when appropriate. Also, as will be demonstrated later in this chapter, a path with an is-a intersection is sometimes the only path which makes any sense of the input.

5.2.7 Finding consistency

As the filter processes path13, it is reminded of the retained path0 because of its similarity to path13. Path0 is evaluated against its competing active path, path2, and is found to provide a better explanation of the text than path2. Subsequently, path1 is re-evaluated because of its earlier tie with path0 but is found to be less explanatory than path0. The active interpretation is now path0, path6, and path7, and is entirely consistent in its representation: the embassy is a government agency with the inferred goal of preserving security, and the action of searching for microphones is part of a plan for achieving that goal.

```
New path discovered: path13
  Path from INSECT to EMBASSY
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
Also reconsidering (path0) due to shared nodes with path13
Also reconsidering (path1) due to tie with path0
  Shorter-path metric -- path7 shorter than path13
    De-activating path13
  More-reinforcement metric -- path0 has more shared nodes
    than path2
    De-activating path2
    Activating path0
  More-reinforcement metric -- path0 has more shared nodes
    than path1
    De-activating path1
```

5.2.8 A retained path is suppressed

As processing continues, the filter evaluates several new paths and a few rediscovered paths. The evaluation of these new and rediscovered paths in turn results in the re-evaluation of a number of related retained paths. These evaluations do not cause any changes in ATLAST's interpretation. They do, however, allow the filter reduce its overhead by suppressing or eliminating retained paths. When the filter evaluates the rediscovered path20, it also re-evaluates the related path1. Because this is the sixth time that path1 has been re-evaluated, and because ATLAST's relevant parameter says that a retained path can "lose" a maximum of five evaluations, path1 is suppressed. This means that ATLAST no longer has any memory of this particular path being discovered. If the proposer later finds the same sequence of nodes and links in memory, it will be treated as an entirely new path. The benefit gained by eliminating path1 is that ATLAST's workload is reduced: memory requirements are reduced because ATLAST no longer keeps track of path1, and processing requirements are reduced because fewer retained paths means fewer possible re-evaluations of retained paths. Of course, the trick is to eliminate as many retained paths as possible without eliminating *too* many. If the wrong path is eliminated, ATLAST will be unable to recover from its inference error.

```
New path discovered: path20
  Path from MICROPHONE to EMBASSY
    MICROPHONE is an instance of OBJECT
    OBJECT has the instance SECRET
    SECRET is a role-filler of FIND-SECRETS
    FIND-SECRETS is a plan of A-SECRETS
    A-SECRETS is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
Also reconsidering (path18 path16 path14 path1 path19 path12) due
  to shared nodes with path20
  Shorter-path metric -- path7 shorter than path20
    De-activating path20
  Shorter-path metric -- path7 shorter than path18
    De-activating path18
  Shorter-path metric -- path7 shorter than path16
    De-activating path16
  Shorter-path metric -- path7 shorter than path14
    De-activating path14
  More-reinforcement metric -- path0 has more shared nodes
    than path1
    De-activating path1
    Suppressing path1
  Shorter-path metric -- path7 shorter than path19
    De-activating path19
  Shorter-path metric -- path6 shorter than path12
    De-activating path12
```

5.2.9 Wrapping up the first sentence

The capsulizer reads the period that indicates the end of the first sentence, as well as the prepositional phrase and the verb phrase that the capsulizer was parsing. The capsule that tells the filter about the prepositional phrase also informs the filter that the object of the preposition is ambiguous.

```
Capsulizer:
  Retrieving lexical entry: *PERIOD*
  No nodes will be activated from lexical entry
  Sending capsule: (pphrase (INSECT MICROPHONE))
  End prepositional phrase
  End verb phrase
  End sentence
  Sending capsule: (term)
```

The filter assigns the ambiguous object of the prepositional phrase to the object slot of the event being processed. The filter attempts to resolve the ambiguity by examining the active paths to determine if one of the meanings can be eliminated. In this case, no active paths include the `INSECT` node, so the corresponding meaning for the ambiguous word `bugs` is removed from the object slot of the event frame.

```
Filter:
  Received capsule: (pphrase (INSECT MICROPHONE))
    Filling object slot in event0 with INSECT vs. MICROPHONE

  Received capsule: (term)
  End processing of event0
  Begin processing of event1

  No active paths exist through INSECT
    Removing INSECT from object slot in event0
```

ATLAST has now completed processing of the first sentence of the text. The interpretation for this sentence consists of two parts: three paths in ATLAST's memory network, which result from the semantic analysis and are displayed in Figure 5.2,

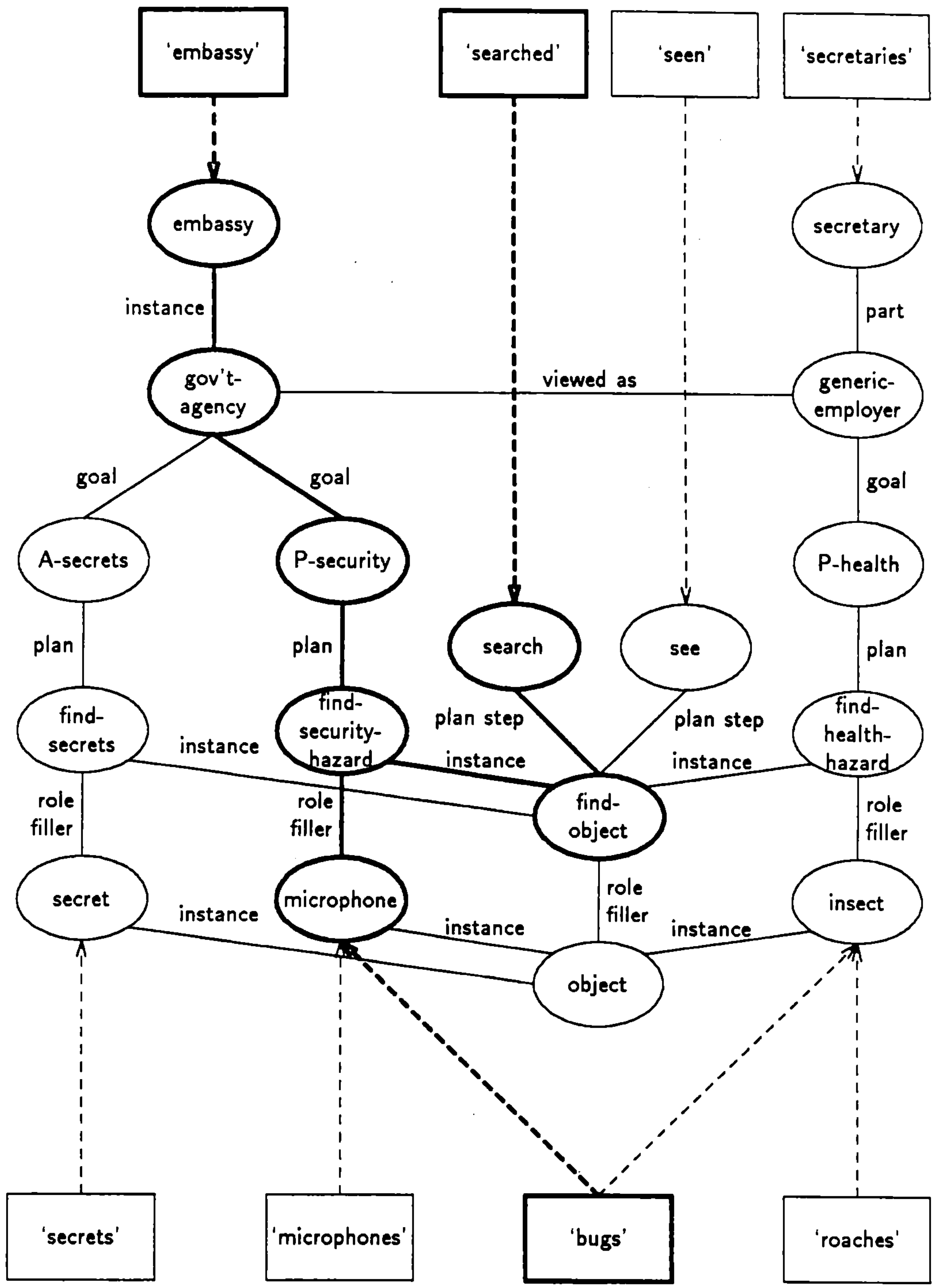

Figure 5.2: Active memory network after processing the first sentence of Text 5.

and the assignment of three of the nodes in those paths to thematic roles, which is done through the syntactic analysis. This independence of semantic and syntactic processing gives ATLAST the ability to process text strings that appear to have meaning but no structure, as well as sentences that have structure but no meaning. This capability will be demonstrated later in this chapter. In order to select the three paths that make up the active interpretation, ATLAST has examined a total of 23 paths, of which 17 have been retained and 3 have been suppressed.

```
Active memory structure:

  Paths: (path0 path6 path7)
  Path from SEARCH to EMBASSY
    SEARCH is a plan step of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
  Path from MICROPHONE to SEARCH
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  Path from MICROPHONE to EMBASSY
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY

Pointers to memory structure:

  Event:  event0
  Actor:  (EMBASSY)
  Action: (SEARCH)
  Object: (MICROPHONE)
  Direction: nil
```

5.3 Intermission

Despite the pages of output, what has happened so far is actually very simple. Essentially, the competition between path7 and path8 was decided by ATLAST's preference for specific over abstract relationships. This decision established a context in which other paths were evaluated, creating an expectation that the embassy's search had more to do with espionage than more mundane, management-related goals. Consequently, the word **bugs** was interpreted as hidden microphones instead of insects.

With regard to what has been demonstrated so far, it is accurate to say that ATLAST is quite similar to other recent NLU systems that employ spreading activation. While the implementation differs greatly, the philosophy remains the same: use spreading activation techniques to search a relational memory network for connections between the words from a text, and then select the connections that best represent the intended meaning of the text. The resolution of ambiguity is simply a by-product of the selection mechanism. However, the selection mechanism may be misled and make incorrect choices. The ability to recover from these errors is what distinguishes ATLAST from other spreading activation systems.

5.4 ATLAST recovers

5.4.1 Beginning the second sentence

The processing of the second sentence is very much like the processing of the first sentence, at least until the end of the sentence. As before, the capsulizer begins by reading the first word, The, which is processed in exactly the same way as it was previously. Because The is a function word, no marker-passing is initiated. ATLAST interprets this word as carrying only syntactic information.[1]

```
Capsulizer:
  Retrieving lexical entry: The
  No nodes will be activated from lexical entry
  Begin sentence
  Begin noun phrase
```

The next word, **secretaries**, initiates marker-passing and several paths are proposed. The filter, however, does not evaluate any of them because of the constraints on what constitutes an acceptable path. Before the filter evaluates a proposed path, the path is examined to see if its endpoints represent words from two different sentences which, as described previously in Chapter 3, ATLAST assumes to represent two different events or states. If the endpoints represent words from a single sentence, the filter evaluates the path. If the endpoints come from different sentences, the path must contain a causal or intentional link, and its endpoints must be nodes representing actions or states if the path is to be evaluated. This constraint enforces the assumption that the primary semantic relationship between two events or states is a causal or intentional relationship (cf. Schank & Abelson, 1977). The paths found by the proposer at this point do not meet this constraint; for example, a proposed path between SECRETARY and SEARCH is not evaluated because SECRETARY is neither an action nor a state. Consequently, the filter is idle and there are no changes to the interpretation.

```
Capsulizer:
  Retrieving lexical entry: secretaries
```

[1]Actually, the word "the" does have semantic content. For example, "the" can denote a particular person, thing, or group that was previously referred to in the text or, as in this case, it can denote something which was not previously mentioned but is not unexpected in the current context. ATLAST is not yet sophisticated enough to handle the semantic subtleties involved here, so it ignores them.

5.4.2 Connecting to the previous sentence

Now the capsulizer encounters the word **had**, which is is syntactically ambiguous. For example, it can be a verb indicating possession or it can be an auxiliary verb used in conjunction with another verb. The former interpretation requires that **had** be treated as an open class word triggering additional marker-passing, while the latter interpretation requires only that it be regarded as a function word. This research concentrates on the resolution of semantic ambiguity, so ATLAST does not know how to resolve syntactic ambiguity. For this example, ATLAST has been instructed to treat **had** as a function word. Meanwhile, the filter assigns the node labeled **SECRETARY** to the actor role in event1.

```
Capsulizer:
  Retrieving lexical entry: had
  No nodes will be activated from lexical entry
  Sending capsule: (nphrase (SECRETARY))
  End noun phrase
  Begin verb phrase

Filter:
  Received capsule: (nphrase (SECRETARY))
    Filling actor slot in event1 with SECRETARY
```

The capsulizer now processes the verb **seen**. Having previously determined that **had** is an auxiliary verb, the capsulizer tells the filter that the sequence of events intended by the text does not correspond to the order in which they have been stated in the text. The capsulizer interprets the construct **had** followed by a verb as an indicator that the event currently being processed actually occurred prior to the event previously processed. The filter postpones changing the temporal order of the chain of events until the processing of event1 is completed. The filter also activates the one path, path23, that connects the two events via intentional links. In this case, the two events, SEE and SEARCH, are interpreted as components of a single plan, FIND-OBJECT.

```
Capsulizer:
  Retrieving lexical entry: seen
    Sending reorder message; current frame is event1

Filter:
  New path discovered: path23
    Path from SEE to SEARCH
      SEE is a plan step of FIND-OBJECT
      FIND-OBJECT has the plan step SEARCH
      Activating path23
```

5.4.3 Continuing with the second sentence

The next two paths represent possible relationships between the words **see** and **secretaries**. The first one, path24, is activated because it is the only path so far that connects these two words. Path24 is quickly de-activated in favor of path25, however, as the latter path fits better with the other active paths.

```
  New path discovered: path24
    Path from SEE to SECRETARY
      SEE is a plan step of FIND-OBJECT
      FIND-OBJECT has the instance FIND-HEALTH-HAZARD
      FIND-HEALTH-HAZARD is a plan of P-HEALTH
      P-HEALTH is a goal of GENERIC-EMPLOYER
      GENERIC-EMPLOYER has the part SECRETARY
  Also reconsidering (path2) due to shared nodes with path24
      Activating path24
    More-reinforcement metric -- path0 has more shared nodes
      than path2
      De-activating path2

  New path discovered: path25
    Path from SEE to SECRETARY
      SEE is a plan step of FIND-OBJECT
      FIND-OBJECT has the instance FIND-SECURITY-HAZARD
```

```
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY can be viewed as GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the part SECRETARY
Also reconsidering (path17 path13 path21) due to shared nodes
  with path25
  More-reinforcement metric -- path25 has more shared nodes
    than path24
    De-activating path24
    Activating path25
  Shorter-path metric -- path7 shorter than path17
    De-activating path17
  Shorter-path metric -- path7 shorter than path13
    De-activating path13
  Shorter-path metric -- path7 shorter than path21
    De-activating path21
```

By the time ATLAST has finished processing **seen**, a total of 27 paths have been discovered and evaluated. ATLAST's active interpretation includes 5 paths and is consistent in its assumption of the espionage theme. Of the remaining 22 paths, 19 are still retained and 3 have been suppressed.

```
Active memory structure:

  Paths: (path25 path23 path0 path6 path7)
  Path from SEE to SECRETARY
    SEE is a plan step of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY can be viewed as GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the part SECRETARY
  Path from SEE to SEARCH
    SEE is a plan step of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  Path from SEARCH to EMBASSY
    SEARCH is a plan step of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
```

```
Path from MICROPHONE to SEARCH
  MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
  FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
  FIND-OBJECT has the plan step SEARCH
Path from MICROPHONE to EMBASSY
  MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
  FIND-SECURITY-HAZARD is a plan of P-SECURITY
  P-SECURITY is a goal of GOVT-AGENCY
  GOVT-AGENCY has the instance EMBASSY
```

5.4.4 A contradiction forces a new interpretation

The last word of the second sentence, **roaches**, is a noun and indicates that the capsulizer is no longer processing a verb phrase. The capsulizer so informs the filter, and the filter assigns the node labeled **SEE** to the action role of the current event frame.

```
Capsulizer:
  Retrieving lexical entry: roaches
  Sending capsule: (vphrase (SEE))
  Begin noun phrase

Filter:
  Received capsule: (vphrase (SEE))
    Filling action slot in event1 with SEE
```

More importantly, the word **roaches** reveals that ATLAST's interpretation of the text is incorrect: the embassy is looking for insects instead of hidden microphones. The marker-passing that is initiated by the proposer in response to this word results in the re-evaluation of several retained paths. The first of these retained paths to be re-evaluated, path3, joins **INSECT** to **SEARCH**. Path3 is compared to its active competitor, path6, which joins **MICROPHONE** to **SEARCH**. Because the node **INSECT** has now been activated by two words from the input text, **bugs** and **roaches**, and the node **MICROPHONE** has been activated only by the word **bugs**, path3 is selected as the better path and path6 is de-activated and retained. The filter then re-evaluates path4, which also joins the nodes INSECT and SEARCH, and is again unable to make a decision between these two paths, so both paths are de-activated and retained.

```
Old path rediscovered: path3
Also reconsidering (path4) due to tie with path3
  More-activation metric -- path3 has more activation than path6
    De-activating path6
    Activating path3
  No-decision metric -- path3 and path4 are retained
    De-activating path3
    De-activating path4
```

The filter evaluates two new paths that connect INSECT to SEE. The first of these, path27, has no active competitor and is immediately activated. The next path, path28, is evaluated against path27, but no decision is made. Both paths are de-activated and retained.

```
New path discovered: path27
  Path from INSECT to SEE
    INSECT is an instance of OBJECT
    OBJECT is a role-filler of FIND-OBJECT
    FIND-OBJECT has the plan step SEE
    Activating path27

New path discovered: path28
  Path from INSECT to SEE
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEE
  No-decision metric -- path27 and path28 are retained
    De-activating path27
    De-activating path28
```

Path29, the first path connecting INSECT to SECRETARY, is discovered and is added to the active interpretation. This new path is related to the retained path8, which is now re-evaluated against the active path7. The filter chooses to replace path7 with path8, again because the latter has INSECT as one of its endpoints while the former has MICROPHONE as its corresponding endpoint.

```
New path discovered: path29
  Path from INSECT to SECRETARY
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is a plan of P-HEALTH
    P-HEALTH is a goal of GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the part SECRETARY
Also reconsidering (path24 path8) due to shared nodes with path29
    Activating path29
  More-reinforcement metric -- path25 has more shared nodes
    than path24
    De-activating path24
  More-activation metric -- path8 has more activation than path7
    De-activating path7
    Activating path8
```

The filter rediscovers path28, which earlier had been de-activated and retained in the tie with path27. Because it has no currently active competing path, path28 is activated. Path27 is again re-evaluated against path28, but the activation of path29 has changed the context enough so that path28 is favored over path27 in the competition.

```
Old path rediscovered: path28
Also reconsidering (path27) due to tie with path28
    Activating path28
  More-reinforcement metric -- path28 has more shared nodes
    than path27
    De-activating path27
```

Path3 is discovered again and compared to path4 because of an earlier tie between the two paths. This time, the context has changed sufficiently so that a decision is made: path4 fits better with the existing context and is chosen over path3, which is now suppressed.

```
Old path rediscovered: path3
Also reconsidering (path4) due to tie with path3
    Activating path3
  More-reinforcement metric -- path4 has more shared nodes
    than path3
    De-activating path3
    Suppressing path3
    Activating path4
```

A few new paths are discovered and evaluated, and a number of retained paths are re-evaluated, but there are no changes to the interpretation. This activity does result in more retained paths being suppressed. The last new path to be discovered, path32, is not incorporated into ATLAST's interpretation. On the other hand, path32's two related retained paths, path24 and path2, are re-evaluated and replace their competing paths, path25 and path0 respectively because they support the interpretation of **bugs** as insects.

```
New path discovered: path32
  Path from INSECT to SECRETARY
    INSECT is an instance of OBJECT
    OBJECT is a role-filler of FIND-OBJECT
    FIND-OBJECT has the instance FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is a plan of P-HEALTH
    P-HEALTH is a goal of GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the part SECRETARY
Also reconsidering (path24 path2) due to shared nodes with path32
  Shorter-path metric -- path29 shorter than path32
    De-activating path32
  More-reinforcement metric -- path24 has more shared nodes
    than path25
    De-activating path25
    Activating path24
  More-reinforcement metric -- path2 has more shared nodes
    than path0
    De-activating path0
    Activating path2
```

Again examining the active paths, the filter determines that its original assignment of the `MICROPHONE` node to the object role in the previous event frame is incorrect because several of the active paths include the INSECT node but no active paths include the MICROPHONE node. To correct its mistake, the filter now assigns INSECT to the object role.

```
No active paths exist through MICROPHONE
  Removing MICROPHONE from object slot in event0

Active paths exist through INSECT again
  Adding INSECT to object slot in event0
```

5.4.5 The light at the end of the tunnel

Finally, ATLAST finds the end of the text. The capsulizer tells the filter that it has finished processing both the noun phrase, the verb phrase in which it was embedded, and the sentence.

```
Capsulizer:
  Retrieving lexical entry: *PERIOD*
  No nodes will be activated from lexical entry
  Sending capsule: (nphrase (INSECT))
  End noun phrase
  End verb phrase
  End sentence
  Sending capsule: (term)
```

The filter assigns the node INSECT to the object role in event1, then reverses the temporal order of event0 and event1 as discussed previously.

```
Filter:
  Received capsule: (nphrase (INSECT))
    Filling object slot in event1 with INSECT

  Received capsule: (term)
  End processing of event1
    Reversing order of last two events
```

ATLAST has now completed its processing of the text. It has found 33 different inference paths and has settled on 7 of them as the best interpretation of the text. These paths are displayed in Figure 5.3. Of the remaining 26 paths, only 16 are retained when processing is completed. Because there is no more text to be processed, all retained paths are suppressed.

```
Suppressing 16 retained paths

Processing completed

  Active memory structure:

    Paths: (path2 path24 path4 path28 path8 path29 path23)
    Path from SEARCH to EMBASSY
      SEARCH is a plan step of FIND-OBJECT
      FIND-OBJECT has the instance FIND-HEALTH-HAZARD
      FIND-HEALTH-HAZARD is a plan of P-HEALTH
      P-HEALTH is a goal of GENERIC-EMPLOYER
      GENERIC-EMPLOYER can be viewed as GOVT-AGENCY
      GOVT-AGENCY has the instance EMBASSY
    Path from SEE to SECRETARY
      SEE is a plan step of FIND-OBJECT
      FIND-OBJECT has the instance FIND-HEALTH-HAZARD
      FIND-HEALTH-HAZARD is a plan of P-HEALTH
      P-HEALTH is a goal of GENERIC-EMPLOYER
      GENERIC-EMPLOYER has the part SECRETARY
    Path from INSECT to SEARCH
      INSECT is a role-filler of FIND-HEALTH-HAZARD
      FIND-HEALTH-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the plan step SEARCH
    Path from INSECT to SEE
      INSECT is a role-filler of FIND-HEALTH-HAZARD
      FIND-HEALTH-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the plan step SEE
    Path from INSECT to EMBASSY
      INSECT is a role-filler of FIND-HEALTH-HAZARD
      FIND-HEALTH-HAZARD is a plan of P-HEALTH
      P-HEALTH is a goal of GENERIC-EMPLOYER
      GENERIC-EMPLOYER can be viewed as GOVT-AGENCY
      GOVT-AGENCY has the instance EMBASSY
```

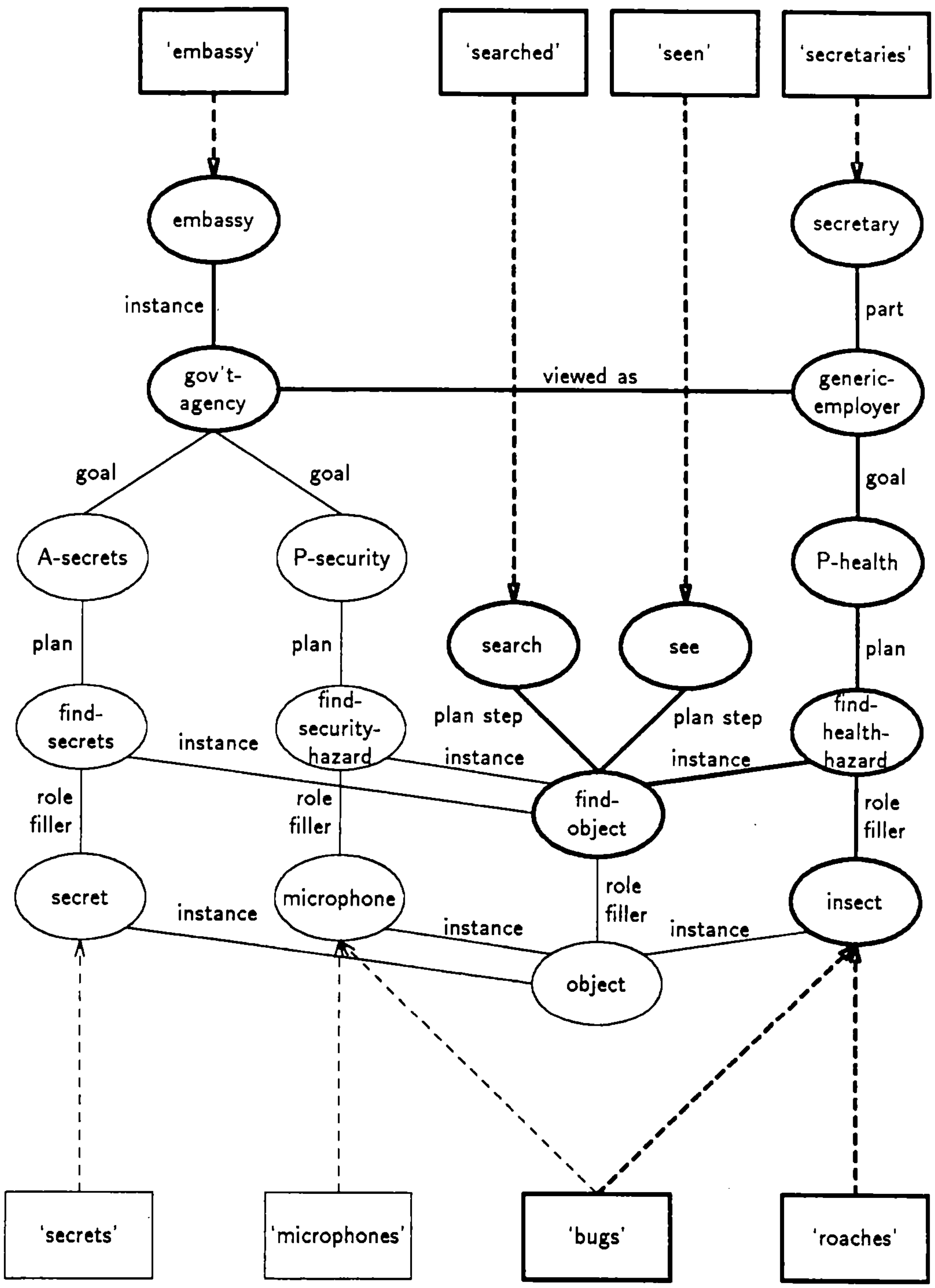

Figure 5.3: Active memory network after processing both sentences of Text 5.

```
Path from INSECT to SECRETARY
  INSECT is a role-filler of FIND-HEALTH-HAZARD
  FIND-HEALTH-HAZARD is a plan of P-HEALTH
  P-HEALTH is a goal of GENERIC-EMPLOYER
  GENERIC-EMPLOYER has the part SECRETARY
Path from SEE to SEARCH
  SEE is a plan step of FIND-OBJECT
  FIND-OBJECT has the plan step SEARCH

Pointers to memory structure:

  Event:   event1
  Actor:   (SECRETARY)
  Action:  (SEE)
  Object:  (INSECT)
  Direction: nil

  Event:   event0
  Actor:   (EMBASSY)
  Action:  (SEARCH)
  Object:  (INSECT)
  Direction: nil
```

5.5 Life without retention

Within the ATLAST framework, conditional retention of inferences is essential to the understanding of misleading texts such as Text 5 without reprocessing. A prediction which follows from this claim is that processing Text 5 without retention should result in an incorrect interpretation. This section summarizes what happens when ATLAST processes Text 5 without the ability to retain rejected inference paths. Here ATLAST is run with the same parameter settings as those used above with the exception that any path which is not activated upon evaluation is immediately suppressed.

When ATLAST processes the first part of the first sentence, `The embassy searched`, the various paths between `SEARCH` and `EMBASSY` are discovered and evaluated. These comparisons result in a series of split decisions as before, but this time the paths are suppressed instead of retained. As ATLAST continues processing the remainder of the first sentence, `for bugs`, it finds the paths joining `EMBASSY` to `INSECT` and `EMBASSY` to `MICROPHONE`. ATLAST again favors the more specific interpretation based on espionage, but the appropriate path between `SEARCH` and `EMBASSY` has not been retained nor is it rediscovered, so it cannot be re-evaluated and added to the interpretation. ATLAST's active interpretation at the end of the first sentence is given below. (The path names are different from those in the previous example because ATLAST's retention ability is disabled. The filter gives a path a new name if it has never seen the path before, and ATLAST now has no recollection of having previously seen any paths other than the currently active paths.)

```
Active memory structure:

  Paths: (path19 path11)
  Path from MICROPHONE to SEARCH
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  Path from MICROPHONE to EMBASSY
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY

Pointers to memory structure:

  Event:  event0
  Actor:  (EMBASSY)
  Action: (SEARCH)
  Object: (MICROPHONE)
  Direction: nil
```

Two additional paths are activated during the processing of The secretaries had seen. One path joins SEE to SEARCH and the other joins SEE to SECRETARY. The interpretation still follows the espionage theme.

```
Active memory structure:

  Paths: (path43 path41 path19 path11)
  Path from SEE to SECRETARY
    SEE is a plan step of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY can be viewed as GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the part SECRETARY
  Path from SEE to SEARCH
    SEE is a plan step of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  Path from MICROPHONE to SEARCH
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  Path from MICROPHONE to EMBASSY
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
```

When it reads the final word of the text, roaches, ATLAST activates a path connecting INSECT to SECRETARY which infers the health-preservation goal of the generic employer. The word roaches makes it clear that the previously inferred goal of preserving security goal is wrong, but ATLAST cannot supplant the older inference paths because the paths which should take their places have been suppressed. Also, ATLAST is unable to activate the correct path between INSECT and SEE because the evaluation of candidate paths resulted in split decisions and suppression of the paths. The inconsistency now inherent in ATLAST's final interpretation is reflected in the two different meanings bound to the object slot of event0.

```
Active memory structure:

  Paths: (path48 path43 path41 path19 path11)
  Path from INSECT to SECRETARY
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is a plan of P-HEALTH
    P-HEALTH is a goal of GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the part SECRETARY
  Path from SEE to SECRETARY
    SEE is a plan step of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY can be viewed as GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the part SECRETARY
  Path from SEE to SEARCH
    SEE is a plan step of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  Path from MICROPHONE to SEARCH
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  Path from MICROPHONE to EMBASSY
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY

Pointers to memory structure:

  Event:  event1
  Actor:  (SECRETARY)
  Action: (SEE)
  Object: (INSECT)
  Direction: nil

  Event:  event0
  Actor:  (EMBASSY)
  Action: (SEARCH)
  Object: (INSECT MICROPHONE)
  Direction: nil
```

In the example immediately above, ATLAST expends less effort and finds a solution more quickly than was the case in the previous example, but is unable to arrive at a correct or even a semantically consistent interpretation. The only difference between the version of ATLAST used here and the earlier one which worked correctly is that the earlier one *retained previously rejected inference paths.* However, conditional retention is only part of the story. The other necessary component to understanding and error recovery in ATLAST is the ability to re-evaluate possibly relevant retained paths at the appropriate times, for without a mechanism for knowing when and how to re-evaluate the retained paths, the retention feature alone provides no benefit.

There are two ways in which the re-evaluation of a retained path can be initiated. The first is through direct rediscovery of the retained path by the search process. Because the passing of markers begins in different places at different times during the processing of text, the same inference path may be discovered (or more appropriately, rediscovered) more than once. If a rediscovered path is not currently part of ATLAST's interpretation of the text (i.e., the path has been discovered earlier, rejected by the evaluation metrics, but retained), that path is re-evaluated against the competing path which is part of the interpretation. This rediscovery process initiates reconsideration of some of the retained paths, but it is not dependent upon retention because these paths would be reconsidered even if they had not been retained.

Some retained paths, though, will not be rediscovered, but the inferences made from later text may change the interpretation in such a way that these paths now should be included. ATLAST uses a method of "piggy-backing" the re-evaluation of these paths onto the evaluation of paths which are directly discovered or rediscovered by the search process. If a (re)discovered path is evaluated against a competing path in the current interpretation, any subpaths or superpaths of the (re)discovered path are also evaluated against the current interpretation. In this way, ATLAST attempts to limit re-evaluation to those paths that are currently relevant. Without the ability to force re-evaluation of paths rejected early in processing but not rediscovered later, ATLAST's final interpretation probably will be incorrect. Indirectly initiating the re-evaluation of previously rejected inference paths is essential to ATLAST's error recovery capability and is dependent upon inference retention.

5.6 The three steps to error recovery

Another way to view error recovery is as a three-step process (Norvig, 1983). The three steps are (1) recognizing that an error has occurred, (2) locating the source of the error, and (3) correcting the error. From this perspective, ATLAST's approach to error recovery can be summarized as follows:

- **Recognizing that an error has occurred:** Each inference path has only two endpoints, and for any two given endpoints there will be at most one active inference path between them. When a new path is discovered (or an old path is rediscovered), the set of currently active paths is searched for a path which shares the same endpoints. If such a path exists, it is possible that the currently active path was incorrectly included in the representation of the text.
- **Locating the source of the error:** This step is effectively subsumed by the previous one. If competition between inference paths has been detected, the competition will always be between a path which is currently part of the representation and one which is not. If an error has in fact occurred, the source of the error will be the currently active path.
- **Correcting the error:** The competing paths are evaluated in the context of the current interpretation of the text (minus the active path being evaluated). The path which is more appropriate to the current interpretation is added to the interpretation, while the less appropriate path is added to the set of retained paths. If the interpretation changed as a result, then an error has been detected, located, and corrected.

5.7 Functional independence in action

Chapter 2 argued for the functional independence of syntax and semantics, based in part on observations that human understanders are able to determine the syntactic correctness of semantically anomalous texts and are also able to assign meaning to agrammatical strings of semantically related words (Chomsky, 1957; Winograd, 1973). The principle of functional independence of syntax and semantics has been maintained during ATLAST's development; accordingly, ATLAST is also able to process texts which are either syntactically or semantically deficient with some degree of success. The following two examples demonstrate ATLAST's ability to process ill-formed texts. ATLAST's inference retention capability has been restored and all other parameters are set to the values given at the beginning of this chapter.

5.7.1 Syntax without semantics

In the first example, ATLAST attempts to understand a sentence which is syntactically valid but has little semantic value. The capsulizer parses the sentence correctly, but the proposer finds no inference paths in memory until the last word of the text is processed.

```
Input text is:

    The lamp proposed to the microphones.

Processing begins

Capsulizer:
  Retrieving lexical entry: *START*
  No nodes will be activated from lexical entry
  Sending capsule: (start)
```

```
Filter:
  Received capsule: (start)
  Begin processing of event0

Capsulizer:
  Retrieving lexical entry: The
  No nodes will be activated from lexical entry
  Begin sentence
  Begin noun phrase

Capsulizer:
  Retrieving lexical entry: lamp

Capsulizer:
  Retrieving lexical entry: proposed
  Sending capsule: (nphrase (LAMP))
  End noun phrase
  Begin verb phrase

Filter:
  Received capsule: (nphrase (LAMP))
    Filling actor slot in event0 with LAMP

  Active memory structure:

    Paths: nil

Capsulizer:
  Retrieving lexical entry: to
  No nodes will be activated from lexical entry
  Sending capsule: (vphrase (PROPOSE-MARRIAGE))
  Begin prepositional phrase

Filter:
  Received capsule: (vphrase (PROPOSE-MARRIAGE))
    Filling action slot in event0 with PROPOSE-MARRIAGE

Capsulizer:
  Retrieving lexical entry: the
  No nodes will be activated from lexical entry

Capsulizer:
  Retrieving lexical entry: microphones
```

The only paths found by the proposer are four paths connecting MICROPHONE and LAMP. The filter activates the shortest path which says simply that a microphone and a lamp are both objects. This is the only semantic content that ATLAST can find in the sentence.

```
Filter:
  New path discovered: path0
    Path from MICROPHONE to LAMP
      MICROPHONE is an instance of OBJECT
      OBJECT has the instance LAMP
      Activating path0

  New path discovered: path1
    Path from MICROPHONE to LAMP
      MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the role-filler OBJECT
      OBJECT has the instance LAMP
    Shorter-path metric -- path0 shorter than path1
      De-activating path1

  New path discovered: path2
    Path from MICROPHONE to LAMP
      MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the instance FIND-HEALTH-HAZARD
      FIND-HEALTH-HAZARD has the role-filler INSECT
      INSECT is an instance of OBJECT
      OBJECT has the instance LAMP
    Shorter-path metric -- path0 shorter than path2
      De-activating path2

  New path discovered: path3
    Path from MICROPHONE to LAMP
      MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the instance FIND-SECRETS
      FIND-SECRETS has the role-filler SECRET
      SECRET is an instance of OBJECT
      OBJECT has the instance LAMP
    Shorter-path metric -- path0 shorter than path3
      De-activating path3
```

Meanwhile, ATLAST continues its syntactic processing with the capsulizer breaking down the sentence into its constituents and the filter binding thematic roles to nodes in the memory. Although ATLAST can find little semantic content, it is able to parse the sentence and build a complete event frame, thus indicating syntactic correctness.

```
Capsulizer:
  Retrieving lexical entry: *PERIOD*
  No nodes will be activated from lexical entry
  Sending capsule: (pphrase-dir (MICROPHONE))
  End prepositional phrase
  End verb phrase
  End sentence
  Sending capsule: (term)

Filter:
  Received capsule: (pphrase-dir (MICROPHONE))
    Filling direction slot in event0 with MICROPHONE

  Received capsule: (term)
  End processing of event0

Suppressing 3 retained paths

Processing completed

  Active memory structure:

    Paths: (path0)
    Path from MICROPHONE to LAMP
      MICROPHONE is an instance of OBJECT
      OBJECT has the instance LAMP

  Pointers to memory structure:

    Event:  event0
    Actor:  (LAMP)
    Action: (PROPOSE-MARRIAGE)
    Object: nil
    Direction: (MICROPHONE)
```

All paths found by ATLAST in this example, including the one in the final interpretation, contain is-a intersections. As stated earlier, ATLAST's evaluation metrics will usually prevent a path containing an is-a intersection from appearing in the final interpretation. In this case, however, path0 represents the only explanation that makes any, albeit little, sense to ATLAST. Under similar circumstances it would not be surprising for a human reader to arrive at an interpretation similar to that suggested by path0.[2] It would be surprising, however, for that same reader to arrive at an interpretation along the lines of either path1, path2, or path3, which describe more tenuous and convoluted relationships between a lamp and a microphone than does path0.

5.7.2 Semantics without syntax

The next example shows ATLAST working on a semantically-related set of words which has no sense of syntactic correctness. The words are the open class words of the first sentence of Text 5; hence ATLAST will find the same inference paths it found in processing that same first sentence. This time, however, there are no syntactic cues to tell ATLAST who did what. Because the text is incomplete, ATLAST's syntactic analysis is lamentable but its semantic analysis is not impaired.

```
Input text is:

    search embassy bugs.

Processing begins

Capsulizer:
  Retrieving lexical entry: *START*
```

[2]It might be argued that a human reader would be more likely to come up with metaphorical meanings for a lamp and a microphone, as in the nursery rhyme when the dish ran away with the spoon. This is well beyond ATLAST's abilities.

```
  No nodes will be activated from lexical entry
  Sending capsule: (start)

Filter:
  Received capsule: (start)
  Begin processing of event0
```

The first word of the input string, `search`, is a verb, so the capsulizer assumes the beginning of a verb phrase.

```
Capsulizer:
  Retrieving lexical entry: search
  Begin sentence
  Begin verb phrase
```

The next word is a noun, so the capsulizer notifies the filter that `SEARCH` is the action in the event frame it is building. The capsulizer then begins processing a noun phrase embedded in a verb phrase. The proposer discovers three paths joining `EMBASSY` and `SEARCH` and activates one of them.

```
Capsulizer:
  Retrieving lexical entry: embassy
  Sending capsule: (vphrase (SEARCH))
  Begin noun phrase

Filter:
  Received capsule: (vphrase (SEARCH))
    Filling action slot in event0 with SEARCH

  New path discovered: path0
    Path from EMBASSY to SEARCH
      EMBASSY is an instance of GOVT-AGENCY
      GOVT-AGENCY has the goal A-SECRETS
      A-SECRETS has the plan FIND-SECRETS
      FIND-SECRETS is an instance of FIND-OBJECT
      FIND-OBJECT has the plan step SEARCH
      Activating path0

  New path discovered: path1
    Path from EMBASSY to SEARCH
```

```
    EMBASSY is an instance of GOVT-AGENCY
    GOVT-AGENCY has the goal P-SECURITY
    P-SECURITY has the plan FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
  No-decision metric -- path0 and path1 are retained
    De-activating path0
    De-activating path1

New path discovered: path2
  Path from EMBASSY to SEARCH
    EMBASSY is an instance of GOVT-AGENCY
    GOVT-AGENCY can be viewed as GENERIC-EMPLOYER
    GENERIC-EMPLOYER has the goal P-HEALTH
    P-HEALTH has the plan FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
    Activating path2
```

ATLAST reads the next word of the text, **bugs**, which the capsulizer treats as part of the noun phrase it is processing. The processing of **bugs** generates 20 new inference paths, but only a few of these will affect the final interpretation.

```
Capsulizer:
  Retrieving lexical entry: bugs
  Ambiguous word senses noted: (INSECT MICROPHONE)

Filter:
  New path discovered: path5
    Path from MICROPHONE to SEARCH
      MICROPHONE is an instance of OBJECT
      OBJECT is a role-filler of FIND-OBJECT
      FIND-OBJECT has the plan step SEARCH
      Activating path5

  New path discovered: path6
    Path from MICROPHONE to SEARCH
      MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the plan step SEARCH
```

```
  No-decision metric -- path5 and path6 are retained
    De-activating path5
    De-activating path6

New path discovered: path7
  Path from MICROPHONE to EMBASSY
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
Also reconsidering (path1) due to shared nodes with path7
Also reconsidering (path0) due to tie with path1
    Activating path7
  More-specific metric -- path1 more specific than path2
    De-activating path2
    Activating path1
  No-decision metric -- path1 and path0 are retained
    De-activating path1
    De-activating path0

New path discovered: path8
  Path from INSECT to EMBASSY
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is a plan of P-HEALTH
    P-HEALTH is a goal of GENERIC-EMPLOYER
    GENERIC-EMPLOYER can be viewed as GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
Also reconsidering (path2) due to shared nodes with path8
  More-specific metric -- path7 more specific than path8
    De-activating path8
    Activating path2

New path discovered: path9
  Path from INSECT to SEARCH
    INSECT is an instance of OBJECT
    OBJECT has the instance MICROPHONE
    MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the plan step SEARCH
Also reconsidering (path6) due to shared nodes with path9
Also reconsidering (path5) due to tie with path6
    Activating path9
```

```
Shorter-path metric -- path6 shorter than path9
  De-activating path9
  Activating path6
More-reinforcement metric -- path6 has more shared nodes
  than path5
  De-activating path5
```

At this point the final interpretation includes path2, path6, and path7 and is semantically inconsistent. When the filter evaluates path13, it supplants path2 with path1 and the interpretation becomes consistent in its assumption of the espionage theme.

```
New path discovered: path13
  Path from INSECT to EMBASSY
    INSECT is a role-filler of FIND-HEALTH-HAZARD
    FIND-HEALTH-HAZARD is an instance of FIND-OBJECT
    FIND-OBJECT has the instance FIND-SECURITY-HAZARD
    FIND-SECURITY-HAZARD is a plan of P-SECURITY
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY
Also reconsidering (path1) due to shared nodes with path13
Also reconsidering (path0) due to tie with path1
  Shorter-path metric -- path7 shorter than path13
    De-activating path13
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2
    Activating path1
  More-reinforcement metric -- path1 has more shared nodes
    than path0
    De-activating path0
```

Reaching the end of the text string, the capsulizer notes the end of the noun phrase, the verb phrase in which it is embedded, and the sentence. The capsulizer assumes that the last noun in the noun phrase, `bugs`, is the head noun. The other noun, `embassy`, is treated as a modifier.

```
Capsulizer:
  Retrieving lexical entry: *PERIOD*
```

```
No nodes will be activated from lexical entry
Sending capsule: (nphrase (INSECT MICROPHONE) (EMBASSY))
End noun phrase
End verb phrase
End sentence
Sending capsule: (term)
```

The filter receives the capsule indicating the noun phrase and fills the object role of the event frame with pointers to the nodes activated by bugs. Unfortunately, the filter has no idea what to do with modifiers, so the pointer to the node EMBASSY is ignored. Even if it did know about modifiers, the filter would still not entertain the possibility that EMBASSY might be a better actor than modifier. On a more positive note, ATLAST does resolve the ambiguity between INSECT and MICROPHONE.

```
Filter:
  Received capsule: (nphrase (INSECT MICROPHONE) (EMBASSY))
    Filling object slot in event0 with INSECT vs. MICROPHONE

  Received capsule: (term)
  End processing of event0

  No active paths exist through INSECT
    Removing INSECT from object slot in event0

  Active memory structure:

    Paths: (path1 path6 path7)
    Path from EMBASSY to SEARCH
      EMBASSY is an instance of GOVT-AGENCY
      GOVT-AGENCY has the goal P-SECURITY
      P-SECURITY has the plan FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the plan step SEARCH
    Path from MICROPHONE to SEARCH
      MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is an instance of FIND-OBJECT
      FIND-OBJECT has the plan step SEARCH
    Path from MICROPHONE to EMBASSY
      MICROPHONE is a role-filler of FIND-SECURITY-HAZARD
      FIND-SECURITY-HAZARD is a plan of P-SECURITY
```

```
    P-SECURITY is a goal of GOVT-AGENCY
    GOVT-AGENCY has the instance EMBASSY

Pointers to memory structure:

  Event:  event0
  Actor:  nil
  Action: (SEARCH)
  Object: (MICROPHONE)
  Direction: nil
```

In summary, ATLAST has found the same active memory structure in this example as it found after processing the first sentence of Text 5 at the beginning of this chapter. Its syntactic processors have made the best analysis they could based on what limited knowledge they have, but the important point is that neither the text's agrammaticality nor ATLAST's weak syntactic processors prevented ATLAST from finding semantic content in the text string. Similarly, the lack of semantic content in the text used in the previous example did not prevent ATLAST from correctly parsing that text. Very few computational models of language understanding are able to demonstrate these human language processing characteristics.

Chapter 6

Pragmatic Inference Processing: Theory

6.1 The problem of pragmatic inference

As defined by Seifert, Robertson, and Black (1982), *pragmatic inferences* are connections between propositions conveyed by a text and world knowledge previously stored in memory (see also Abelson & Reich, 1969).[1] These inferences are plausible but are not necessarily true. Applying this definition to the ATLAST framework, a pragmatic inference is a decision about which inference path best represents an implicit relationship between events or states explicitly given in a text.

At any point in the text where a pragmatic inference could be made, there is usually more than one inference possible. However, in explaining the events in a text, some inferences may serve better than others. For example, after reading the following text,

[1]This definition is related to, but not necessarily the same as, the usage of the word *pragmatics* in psycholinguistics. For many psycholinguists, the term *pragmatics* refers to the language user's knowledge of the *social* rules underlying language use (Carroll, 1986). While the social rules are part of a language user's world knowledge, many classes of inferences one can make using such rules are beyond the scope of this dissertation.

Text 6: John was poor but he owned a gun.
He went to the pawnshop.

the reader might infer that John intended to rob the pawnshop. On the other hand, John might be intending to get money for his gun at the pawnshop. Either interpretation explains the explicit text events, although individual readers may ascribe different degrees of plausibility to the two interpretations.[2] As is the case with lexical inference processing, the primary problem in pragmatic inference processing is disambiguation: the evaluation of competing inferences and subsequent selection of the one which best explains the events portrayed in the text.

In the example above, the pragmatic inferences that can be made do not hinge on decisions about the meanings of the individual words that are contained in Text 6. The words "poor," "gun," "pawnshop," and so on carry the same meaning in this context regardless of which of the two interpretations is chosen. The ambiguity arises because of the different plausible connections the reader can find between the two sentences using his or her knowledge of how the world works. In terms of the perspective described by Schank and Abelson (1977), Text 6 presents the reader with states or conditions (lacking money and owning a gun) and an action (going to the pawn shop). The reader then infers a corresponding goal (get money). There are a number of plans that can be inferred for achieving the goal; at least two are suggested by the action of taking the gun to the pawnshop (steal money or sell gun). To resolve the ambiguity, the reader must somehow evaluate the explanatory value of the two plans and select the one that seems more plausible.[3]

[2]Wilensky (1978) uses a similar story: "John wanted money. He got a gun and walked into a liquor store. He told the owner he wanted some money. The owner gave John the money and John left." This story differs from Text 6 in that the only likely use for a gun in the liquor store scenario is to use it to rob the liquor store. While it is undoubtedly possible that John might want to sell or trade the gun at the liquor store, it is not very plausible.

[3]A more accurate and complex analysis of Text 6 is certainly possible, but this will suffice for the purpose at hand.

Let us say for the sake of argument that our hypothetical reader's inference processing mechanism decides that the "steal money" plan is more plausible. The possibility always exists that the reader's selection of the more plausible plan is incorrect, and we might well wonder how that same inference processing mechanism would resolve its decision with the presentation of explicit contradictory information as in Text 7:

Text 7: John was poor but he owned a gun.
He went to the pawnshop.
He sold the gun.

As it was with the theory of lexical inference processing, a theory of pragmatic inference processing should address not only how competing inferences are evaluated and one is selected, but also how selected inferences later may be determined to be incorrect and replaced with more appropriate ones.

That lexical and pragmatic inferences pose common problems suggests that the problems may be answered by a common solution. One possible solution is the topic of this chapter.

6.2 Unifying lexical and pragmatic processing

There is no widespread agreement on the relationship between *word* knowledge and *world* knowledge. For example, participants at one recent workshop represented both sides of the issue: some argued that lexical and pragmatic knowledge are the same (Hobbs, 1987), or at least inextricably bound together (Wilks, 1987), while others argued that the two types of knowledge are quite different and should be treated as such (Israel, 1987; Kegl, 1987). The position taken here draws from both arguments. Knowledge about words and knowledge about the world are inseparable in that word meanings are pointers to relevant parts of the model's pragmatic knowledge, and pragmatic knowledge in turn points to associated word meanings. However, the fact that ATLAST's intra-phrasal syntactic processor makes decisions strictly on the basis of the lexical information presented to it (i.e., the syntactic categories associated with the individual words and the order in which the words are presented) dictates that knowledge about syntactic categories is stored with the lexical entries.[4] Thus, there is a syntactic component to ATLAST's lexical knowledge which is not included in its pragmatic knowledge, thereby distinguishing one from the other, but that difference does not imply that the semantic information in the lexical and pragmatic knowledge levels should be treated differently. In fact, from a semantic processing perspective, ATLAST treats the two levels as one.

Consider again the processing of Text 5:

Text 5: The embassy checked for bugs.
The secretaries had seen roaches.

In this example, competing paths start at one node (`EMBASSY`) but end at two different nodes (`INSECT` and `MICROPHONE`) activated by one word. Early in the text, the inferred goal of the embassy influenced the choice of one path and the associated word meaning, but later text forced a change in word meaning and a shift in the inferred goal of the embassy. In other words, calling up information about the embassy at the lexical level also caused ATLAST to infer goals for the embassy at the pragmatic level. The pragmatic knowledge then influenced ATLAST's choice of meaning for the word `bugs`. Later, the word `roaches` revealed that ATLAST's initial

[4] An experiment by Tyler and Marslen-Wilson (1977) indicates that syntactic decisions also may be influenced by the current semantic context. For reasons of simplicity, ATLAST does not model this behavior, but there is nothing inherent in ATLAST's design which precludes this behavior from being incorporated in the future. Remember that the assumption of functional dependence requires only that syntax and semantics *can* work independently, not that they *must*.

choice of word sense was wrong, and the system revised its inferred goals for the embassy to find the most parsimonious interpretation of the story.

In the case of Text 6, on the other hand, we see no lexical ambiguity. What we do notice is pragmatic ambiguity: is John going to sell his gun or use it to rob the pawnshop? The competing inferences which represent the different interpretations join the same word senses but follow different routes. Thus in one case our attention is drawn to lexical ambiguity, in another case we focus on pragmatic ambiguity, but in either case a decision is made by choosing from competing inferences. In other words, the difference between lexical and pragmatic inferences is not so much in the information in the corresponding paths: both types of paths may contain word senses, related actions, plans, goals, and so on, thus no path can be regarded as purely lexical or purely pragmatic. The difference is in the *structure* of the ambiguity as indicated by the endpoints. If different meanings of the same word are the endpoints of two competing paths, the decision appears to be lexical; without competing word senses at the endpoints, the decision appears to be pragmatic (see Figure 6.1). In either case, deciding between individual paths is fundamentally the same, so a single decision process can be used to make inferences at what are often considered to be different levels of processing.

6.3 Retention of pragmatic inferences

Recall that according to the conditional retention theory, all meanings of an ambiguous word are accessed when the word is processed. The individual meanings are then evaluated in light of the existing context. If one meaning is more appropriate to the context than the others and no text follows, the less appropriate meanings are

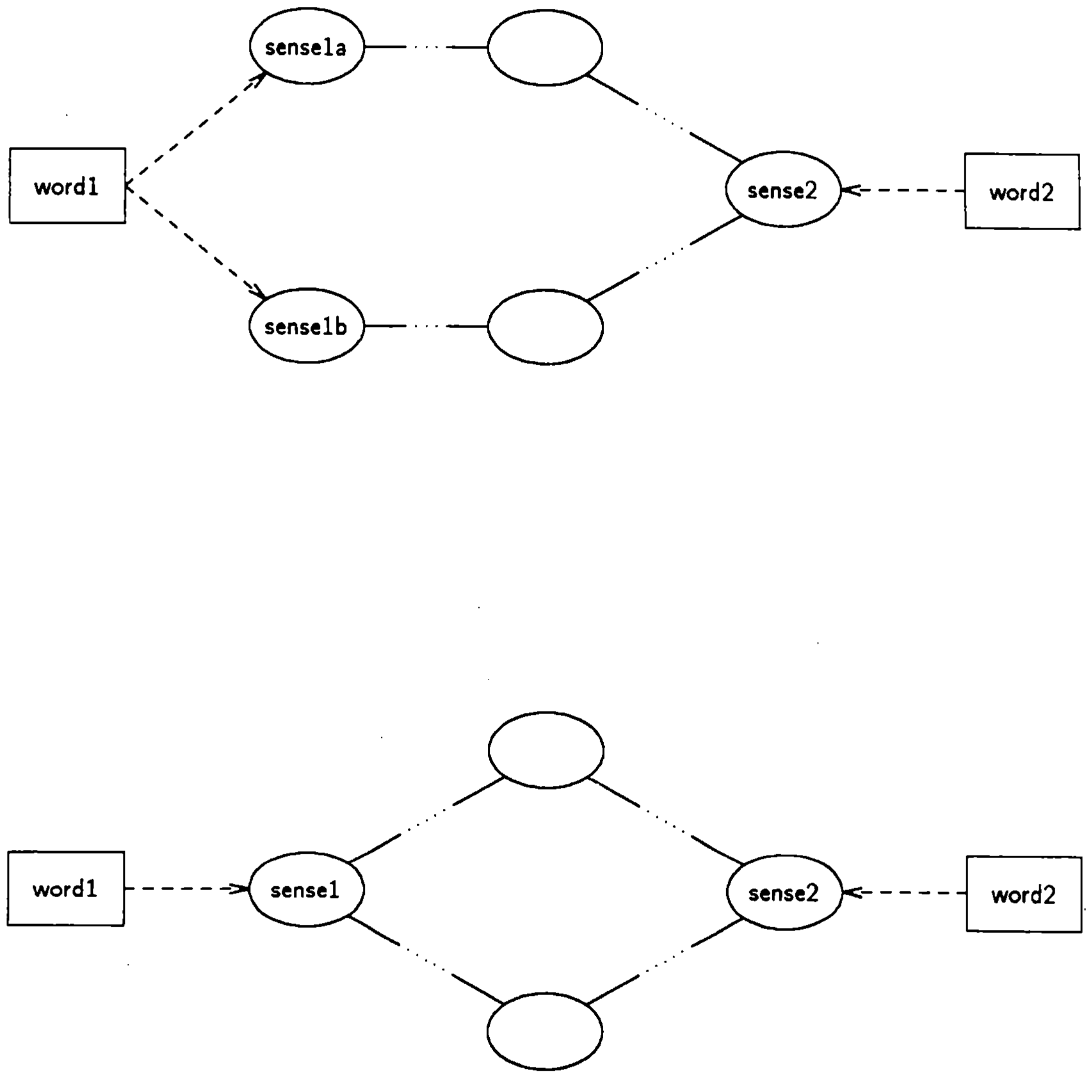

Figure 6.1: The generic structures of lexical ambiguities (above) and pragmatic ambiguities (below).

actively suppressed. If no meaning is preferred by the existing context, or if one is preferred but text follows the ambiguous word, the unchosen meanings are retained until more text can be processed and a decision can be made.

Because ATLAST does not distinguish between paths representing lexical inferences and those that represent pragmatic inferences, the theory of conditional retention also applies to pragmatic inferences. As a result ATLAST can recover, without reprocessing the text, from erroneous pragmatic inference decisions (e.g., Text 7) in the same way it recovers from erroneous lexical inference decisions (e.g., Text 5). This leads to a uniform description of the inference decision and error recovery process which does not distinguish between lexical and pragmatic knowledge:

- Potential inferences which explain the same input text are generated and evaluated in parallel.
- If one inference explains the text better than the others do, and there are no textual cues to indicate that a decision may be premature (e.g., there is still more text to process), then the other less-explanatory inferences are suppressed.
- If no inference proves to be more explanatory than the others, all inferences are retained until later text provides the information necessary to make a decision.

An example of the theory of conditional retention applied to pragmatic inference processing is given in Appendix A, which contains output generated by ATLAST during its processing of a simplified version of Text 7.

The extension of conditional retention theory to pragmatic inference processing enhances ATLAST's appeal as a model of human language understanding because it provides a single, simple mechanism for making inferences and correcting errors at two seemingly different levels of processing. In addition, conditional retention of pragmatic inferences permits a new and interesting perspective on what previously has been called *strategy-driven inference behavior.*

6.4 Strategy-driven inference decisions

An experiment by Granger and Holbrook (1983) revealed that certain texts could induce different readers to arrive at different interpretations of the same story. For example, experimental subjects were presented with the following text:

Text 8: Wilma began to cry.
Fred had just asked her to marry him.

When asked why Wilma cried, some subjects replied that Wilma was unhappy or upset about Fred's marriage proposal, while others answered that Wilma was happy about the proposal and was crying tears of joy. The texts which induced equally plausible and nearly opposite interpretations were called *reciprocally ambiguous* texts. Through a supplementary experiment, Granger and Holbrook eliminated the possibility that different groups of subjects were influenced by contradictory default explanations for the individual story events; in other words, it was not the case that one group thought that a marriage proposal was a happy event and the other considered it to be an unhappy event. When presented with the individual sentences in isolation, subjects would uniformly conclude that Wilma was crying because she was sad or that Fred's marriage proposal was a happy occasion.

Granger and Holbrook theorized that the difference in subjects' interpretations of reciprocally ambiguous stories was due to the existence of different but consistent *strategies* for systematically choosing between competing inference paths. They also theorized the existence of a variety of different strategies and described two strategies in detail.

One proposed strategy was to make inference decisions as early as possible in the reading and then cling to that interpretation as long as possible, fitting new story events into the original interpretation. These readers were called *perseverers.* Another strategy was to postpone decisions until later in the processing of the text. Any conflict in the interpretation would be decided in favor of the most recently presented information. These readers were called *recencies.* Everyday texts tend to be constructed so that any reader would arrive at the same interpretation, regardless of the strategy that was applied, so differences in behavior would likely go unnoticed unless one were specifically looking for such differences.

Individual differences in inference behavior have been addressed, if only briefly, by other researchers. In several previous instances, differences in inference behavior have been attributed to randomness (Hirst, 1988a), subjects' idiosyncracies (Schank, Collins, Davis, Johnson, Lytinen, & Reiser, 1982), or careless reading (Rumelhart, 1981). Granger and Holbrook's data indicate that, at least in this case, the differences are far more systematic and predictable than other investigators might suppose.

A computational model of the processes responsible for differences in inferential behavior was developed soon thereafter (Granger, Eiselt, & Holbrook, 1983). This model, called STRATEGIST, arrived at either of two interpretations of an input text using the same component processes for making pragmatic inferences but different rules for deciding when the processes were invoked (i.e., make the inference now or make the inference later), resulting in different interpretations of the same text. ATLAST followed STRATEGIST as an attempt to answer questions about disambiguation and error recovery that STRATEGIST left unanswered, but those unanswered questions appeared to have relatively little to do with the difference between perseverers and recencies. As work on ATLAST progressed, however, it became apparent that the new architecture could provide valuable insight into strategy-driven inferencing.

Within the ATLAST framework, strategy-driven inference behavior is still explained as a difference between making inference decisions early in processing or postponing them, but the different strategies are invoked only under very specific circumstances. As described previously, ATLAST uses a set of evaluation metrics to choose between two competing inference paths. In most cases, one of the metrics will decide in favor of one path, but occasionally the metrics find no clear winner. As might be expected, path competitions that end in a tie occur more frequently in texts which have two equally plausible and nearly opposite interpretations (i.e., reciprocally ambiguous texts). The difference between perseverers and recencies is how they resolve split decisions.

When a tie between competing paths occurs, both perseverers and recencies will retain the competing paths without making a decision and continue processing the input text. In the case of perseverers, those retained paths are immediately available for re-evaluation should later text indicate that re-evaluation is appropriate. Therefore, although a decision cannot immediately be made, the perseverer is always ready to make that decision as soon as evidence becomes available which favors one path over the other. (In the examples of Chapter 5, ATLAST uses the perseverer strategy.) Recencies, however, will not re-evaluate those split decisions until much later in the processing, ignoring information which possibly could enable an earlier decision. In ATLAST, this is done by preventing re-evaluation on those paths and retaining them until the end of the text, at which point they will be available for re-evaluation. This postponement applies only to the inferences involved in the split decision; all other evaluations result in one inference being activated and the other being retained, as is the case with perseverers. Consequently, when processing texts that generate no split decisions, perseverers and recencies will arrive at the same interpretations. If the text is ambiguous but biased toward one interpretation, the differences in processing should not be obvious: the perseverer may resolve a tie incorrectly early in the processing and then supplant as processing proceeds, while the recency will delay the decision but make the correct one the first time. The final interpretation is the same in either situation. Only rarely, when the equally-plausible competing inferences are fundamental to understanding the text and the choices result in substantially different interpretations, as in Text 8, will the differences be conspicuous.

To aid in the interpretation of reciprocally ambiguous texts, ATLAST uses a sixth and previously unmentioned inference evaluation metric called the *preferred-link metric*. Given two competing inference paths and no winner after the application of the activation, length, reinforcement, and specificity metrics, the preferred-link metric selects the path which contains the greater number of **preferred** links. A preferred link is constructed in ATLAST's memory by adding the **preferred** designator to any of ATLAST's link labels. Thus, **has-result** becomes **has-preferred-result**. The **preferred** link is used to give ATLAST a default interpretation to use when all else fails. For example, if ATLAST were presented with only the first sentence of Text 8, "Wilma began to cry," the system would be unable to determine if Wilma is happy or sad without some built-in predisposition toward one interpretation or the other. Accordingly, a **preferred** link has been used in constructing ATLAST's memory so that, in the absence of any other information, it will favor an inference path which explains crying as a result of sadness over one which explains crying as a result of happiness.

When modeling perseverer behavior, ATLAST adds the preferred-link metric to its list of evaluation metrics before processing begins. The metric is then available for use during the remainder of text processing. The availability of this metric at the beginning of a text means that perseverers use default inferences early in text processing to establish a context for interpreting the remainder of the text. When modeling recency behavior, conversely, ATLAST does not use the preferred-link metric until the end of the text has been processed and the retained split decisions are ready to be re-evaluated. The recency, therefore, first uses default inferences at the end of a text to resolve the postponed split decisions from the more recently processed text, and then uses that context to resolve postponed split decisions from earlier text.

In summary, the difference between the two different inference strategies is nothing more than a difference in how the inference processor deals with ties or split decisions. The perseverer can make use of default inferences early and tries to resolve the ties immediately while the recency cannot use default inferences until the end of the text and postpones the resolution of ties until that time. Any inference paths not involved in split decisions are processed in the same way regardless of the inference strategy in force. It is only this simple difference in processing split decisions that results in the two entirely different interpretations of Text 8. Examples of ATLAST's output while processing a simplified version of Text 8 using both the perseverer and recency strategies are given in Appendix A.

ATLAST thus accounts for one aspect of strategy-driven inference behavior, but what about other phenomena found by Granger and Holbrook that is related to strategy-driven inferencing? For example, Granger and Holbrook found that if a perseverer were presented with a story like Text 8 but with the events given in reverse order, the perseverer would arrive at the interpretation usually attributed to recency behavior. That is, if presented with the following text,

Text 9: Fred had just asked Wilma to marry him.
She began to cry.

the perseverer would conclude that Wilma was crying tears of joy. Likewise, the recency would conclude that Wilma was sad or upset. ATLAST behaves similarly when the order of the story events is reversed. Granger and Holbrook also found that when the individual story events were presented in isolation (i.e., only one of the two sentences was presented), perseverers and recencies arrived at the same interpretation. When ATLAST is presented with only one of the story events, it too arrives at only one interpretation regardless of the inference strategy it is using. Finally, Granger and Holbrook predicted that individual differences in strategy-driven inference behavior would become visible only through the use of specially constructed reciprocally ambiguous stories. Given texts that were not so constructed, recencies and perseverers would arrive at the same interpretation. Again, ATLAST's two different strategies find only one interpretation for stories which are not reciprocally ambiguous, such as Text 5 and Text 7. Thus, ATLAST accounts for a wide variety of phenomena related to strategy-driven inference processing.

Retention of inferences plays an essential role in strategy-driven inferencing. Ignoring the issue of error recovery for the time being, the perseverer does not rely on retention nearly as much as the recency, if at all. The inferences retained due to a perseverer's split decision will be re-evaluated very quickly because of their similarity to other inferences soon to be evaluated. If those inferences had not been retained, they most likely would have been rediscovered by the inference proposer anyway. The recency, on the other hand, requires the retention of those same inferences. As processing of the text proceeds beyond the occurrence of the split decision and into another sentence or two, the locus of activity within the memory changes, and the likelihood of the tied inferences being generated again decreases considerably. If the tied inferences are not retained for later re-evaluation, the recency has little chance of incorporating either of the inferences into its interpretation of the input. In other words, without retention of competing inferences, there can be no postponement of the inference decision, and without postponement there can be no recency behavior.

In ATLAST, the difference between perseverer and recency behavior is controlled by changing a parameter. Flip a switch and retained paths resulting from split decisions cannot be re-evaluated until the end of the text; flip it back and all retained paths can be re-evaluated on demand. Thus, all things remain constant between recency operation and perseverer operation except a change in rules or strategies for dealing with split decisions. Another explanation, which may be more plausible, is that the differences in behavior are caused by minor differences in the underlying cognitive architecture. The behavior might then be more aptly described as *architecture-driven* rather than strategy-driven.

One factor which could account for different interpretations of reciprocally ambiguous texts is a difference in the speed of the spread of activation through memory. A simple prototype model has been built to demonstrate that a connectionist network can be forced to settle into two entirely different activation patterns, each one corresponding to a unique interpretation of the same reciprocally ambiguous text, by increasing or decreasing the individual computing units' sensitivity to activation energy (Eiselt & Granger, 1987). Increasing sensitivity promoted the spread of activation energy through the network and "decisions" were arrived at sooner, resulting in the perseverer interpretation. Conversely, decreasing the sensitivity impeded the spread of activation, decisions were delayed, and the network settled into the recency interpretation.

Still another explanation is possible, one which greatly simplifies the assumptions made in accounting for both error recovery and differences in interpreting text. Assume that all readers share the same "strategy" for resolving split decisions: they retain the competing paths and do not re-evaluate them until the limit on duration of retention is reached. At that time, the inference processor re-evaluates the retained paths in light of whatever context has been built. The only difference between individual readers is in the duration of retention. Thus, readers with short spans will be forced to resolve those ties early and exhibit perseverer behavior. Readers with longer spans will not resolve the ties until much later in processing when more context has been established and will exhibit recency behavior. We might reasonably expect that readers do not fall nicely in to short-retention and long-retention groups; instead, readers might represent a range of different retention spans, which would account for the variety of inference strategies originally predicted by Granger and Holbrook (1983). Again, differences in behavior emerge from variations in the architecture, not from differences in rules.

The explanation just offered finds indirect support in experimental evidence which suggests that reading comprehension is affected by the reader's working memory capacity (Daneman & Carpenter, 1980; Just & Carpenter, 1980; Kintsch & van Dijk, 1978), but these experiments do not directly address the differences between perseverer and recency behavior. Direct support for this explanation may soon be available, however. A preliminary appraisal of experimental data obtained by Holbrook (in preparation) indicates that human subjects exhibit different behaviors in resolving lexical ambiguity. The differences appear to correspond to the relative speed with which the subject makes a decision about an ambiguous word. This observation, yet to be subjected to a thorough analysis of the data as of this writing, is compatible with the explanation of error recovery and inference processing differences offered above, and suggests the existence of perseverer and recency strategies at the lexical processing level.

6.5 Conclusion

The theory of pragmatic inference processing proposed in this chapter provides a simple account of a variety of phenomena including lexical disambiguation, pragmatic disambiguation, detection and correction of errors during disambiguation at both levels, and individual differences in strategy-driven inference behavior. The theory has been extrapolated from experimental evidence pertaining to lexical inference processing but it has not been tested at the pragmatic level, other than with the construction of a working computer program. The latter is only a test of plausibility, however, not of accuracy. Still, ATLAST is the only model which accounts for all these different phenomena. Its versatility is demonstrated in Appendix A through annotated examples of ATLAST's operation when faced with problems of pragmatic ambiguity, recovery from an incorrect pragmatic inference, and a reciprocally ambiguous text.

The possibility that conditional retention accounts for differences in perseverer and recency behavior, phenomena which at first glance seem to be entirely unrelated, adds to the importance of the conditional retention theory. Just as important, though, is the fact that this exploration of the implications of conditional retention is greatly facilitated by the existence of the ATLAST model. This illustrates one of the primary benefits of building computational models of psychological theories: the model enables the researcher to explore extensions or variations of a theory before designing and running new experiments. Without the ATLAST framework it would have been difficult, if at all possible, to investigate possible relationships between conditional retention and pragmatic inference processing.

Chapter 7

ATLAST as Search

7.1 A different perspective

In the world of cognitive modeling, computer programs usually are presented as formalizations of theories about human cognition and demonstrations of the plausibility of those theories. The programs are subsequently evaluated in terms of how well they account for existing experimental data on human behavior and how well they predict the results of future experiments. Until now, this dissertation has concentrated on how ATLAST accounts for human behavior, but this chapter will examine ATLAST's computational performance.

This shift in perspective is prompted by speculation that a marker-passing search for inference paths will be swamped with unimportant paths as the relational memory network becomes very large. For example, Charniak states:

> The problem with marker passing is that it is not obvious if it can do the job of finding important inferences in a very large and interconnected database. Or to be more precise, can it find the important inferences without finding so many unimportant ones that it becomes useless as an attention focusing device? (Charniak, 1986, p. 588)

This chapter examines the assumptions and constraints relevant to ATLAST's search processes and finds that marker-passing can be efficient and should scale up well.

7.2 Searching for inference paths

Perhaps the single most important concept in artificial intelligence is that of search. There are two different searches being performed as ATLAST processes text: the proposer carries out a breadth-first intersection search for individual inference paths in ATLAST's semantic memory, and the filter uses a hill-climbing technique to search for the combination of proposed paths which best explains the input text. In the former case, the structure of memory defines the search space for the proposer. In turn, the paths found by the proposer serve to define a search space for the filter. This section deals with the first of these two components.

7.2.1 A case study

In the example of Chapter 5, ATLAST interpreted the following text using the semantic memory network shown in Figure 7.1 (a duplicate of Figure 5.1):

Text 5: The embassy checked for bugs.
The secretaries had seen roaches.

During the processing of Text 5, the proposer discovered 33 inference paths, each one representing a different relationship between two selected nodes in memory. The different paths and the nodes which they connect are shown in Table 7.1. Table 7.1 shows that there are three paths connecting `SEARCH` to `EMBASSY`, eight connecting

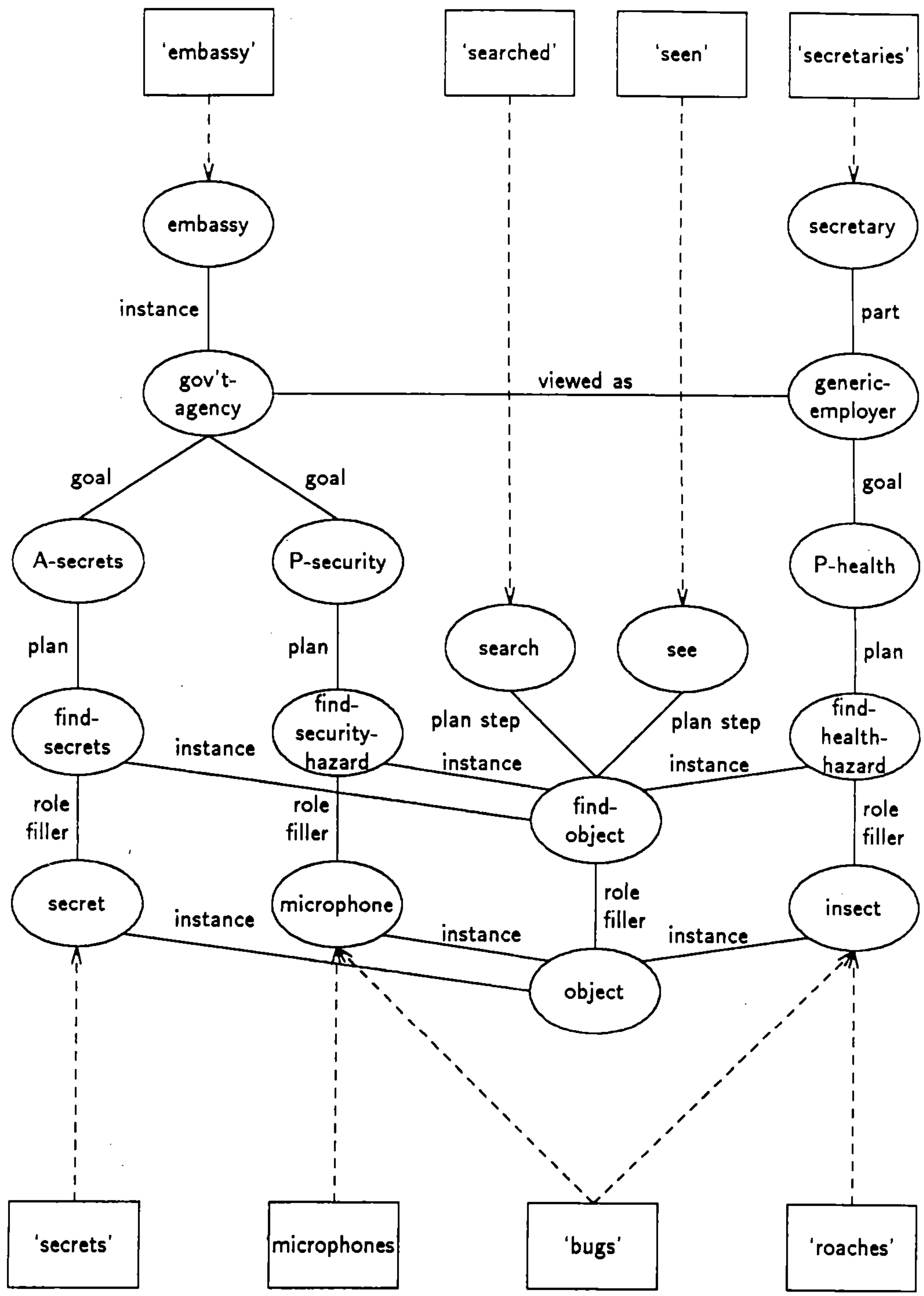

Figure 7.1: Memory network for Text 5.

SEARCH to EMBASSY	INSECT or MICROPHONE to SEARCH	INSECT or MICROPHONE to EMBASSY	SEE to SEARCH	SEE to SECRETARY	INSECT to SEE	INSECT to SECRETARY
path0 path1 path2	path3 path4 path5 path6 path9 path10 path11 path12	path7 path8 path13 path14 path15 path16 path17 path18 path19 path20 path21 path22	path23	path24 path25 path26	path27 path28 path30 path31	path29 path32

Table 7.1: Inference paths discovered during the processing of Text 5.

either `INSECT` or `MICROPHONE` to `SEARCH` (remember that `INSECT` and `MICROPHONE` were activated by the same word, **bugs**), and so on.

The number of paths found by the proposer is constrained by several variables. First, the maximum distance that a marker may travel in memory places a limit on the length of an acceptable path. In the example of Chapter 5, markers could be passed at most three links away from their origin, so the proposer would find no paths longer than twice this distance, six links.[1] This constraint is implemented more elegantly in other systems such as Anderson's ACT* (1983), Hendler's SCRAPS (1986), or Charniak's Wimp (1986), which use a measure of energy instead of distance and exhibit fan-out effects which ATLAST does not. Traversing a link in these systems depletes a marker's energy, and a marker's energy is divided between links when traversing them in parallel. The marker is no longer passed when its energy level drops below a certain point.

Proposed paths must also meet specific criteria regarding the relationships they represent. Paths which include a node more than once (i.e., cycles) are not proposed

[1]This discussion ignores links which effectively have no length as it will only complicate the analysis.

as these tend to be uninformative at best. Also considered to be uninformative are those paths which start and end at competing word senses of the same word. Additionally, while any two origin nodes activated during the processing of a single event may be joined by any combination of links, two origin nodes activated during the processing of different events may be joined only through a causal or intentional relationship between the nodes, and only if the nodes represent either actions or states of the two events. Thus, the proposer generally will find many more intra-event paths than inter-event paths.

The most important constraint is the structure of the semantic memory itself. In the example of Text 5, the semantic memory consists of 17 nodes interconnected by only 21 (of a possible $\binom{17}{2} = 136$) bi-directional links. Six of these nodes were activated directly from the text and serve as the endpoints of the 33 paths found by the proposer. A few nodes exhibit a relatively high degree of interconnectivity: `FIND-OBJECT` is linked to six other nodes, `OBJECT` is linked to four other nodes, and `GOVT-AGENCY` is also linked to four nodes. The remaining nodes are connected to only one, two, or three other nodes, and the average interconnectivity or branching factor for the network is approximately 2.5. Thus, some parts of the network are highly interconnected and other parts are not; the regions of low interconnectivity act as bottlenecks, constraining the spread of markers and limiting the number of new paths found. This is a reasonable and by no means unprecedented assumption about human memory. Although no one is sure how human memory is organized, many models of human memory reflect this assumption, with higher branching factors associated with the organizing features of the particular model, such as goals and plans (Schank & Abelson, 1977), verbs (Kintsch, 1974; Rumelhart & Norman, 1975), or nouns (Kintsch, 1974; Quillian, 1968).

The reader should not infer from this discussion that there is some commonly accepted structural standard for models of human memory. Cognitive scientists build different miniature memory models to illustrate solutions to different problems, and the disparities between these models leave us with no baseline for comparison when trying to determine what is gained by making specific assumptions about memory organization. On the other hand, if we assume that all memory models are roughly equivalent, we can select one sample as representative of the group and compare that sample's optimal behavior to its worst-case behavior.

7.2.2 Some unavoidable mathematics

Using ATLAST's memory for Text 5 as the representative of human memory models, we can examine how assumptions about maximum path length, acceptable relationships between endpoints, and the organization of memory greatly constrain the number of paths in the search space. The worst-case scenario will be the search space generated by a network with the same number of nodes as the representative sample: 17 nodes with each node linked to every other node (a branching factor of 16 at every node). Of course, it is not likely that in human memory every concept would be linked to every other concept, if only because activating one node would remind us of everything else we have stored in memory. Still, this type of network provides a readily accessible baseline for comparison.

For a network of n nodes in which every pair of nodes is joined by a single link (i.e., a completely connected undirected graph on n vertices), the number of paths (P) that join two arbitrary nodes and have length l or less is given by the following equation:[2]

[2]The proof of this theorem is given in Appendix B.

$$P = \sum_{i=1}^{l} \frac{(n-2)!}{(n-i-1)!}$$

In a network of n nodes, the longest path possible without a cycle has length $n-1$, so the longest path in our 17-node network contains 16 links. Using the formula above to compute the number of paths of length 16 or less between any two nodes in this network, we find that P is *very* large:

$$P = \sum_{i=1}^{16} \frac{(17-2)!}{(17-i-1)!} = 3,554,627,472,076$$

Text 5 activated six nodes: `EMBASSY`, `SEARCH`, `INSECT`, `MICROPHONE`, `SECRETARY`, and `SEE`. From these six nodes, $\binom{6}{2}$ or 15 different combinations of nodes can be selected, so without any constraints the proposer would find $15 \times 3,554,627,472,076$ or $53,319,412,081,140$ paths. This is the worst-case scenario.

By imposing a limit of six on the path length, the number of paths that can be found by the proposer is dramatically reduced:

$$15 \times P = 15 \times \sum_{i=1}^{6} \frac{(17-2)!}{(17-i-1)!} = 15 \times 396,076 = 5,941,140$$

In other words, by imposing the constraint on the path length, the number of paths that could be found by the proposer has been reduced by seven orders of magnitude. More important, however, is a general observation based on the formula for P above. If there is no fixed limit on path length (l), P will grow exponentially as n increases. On the other hand, if l is fixed as n increases, P grows only polynomially. Thus, the limit on path length prevents the search problem from being as bad as it could be.

Imposing constraints on the structure of the network so that there is no longer a single link joining each pair of nodes will also further reduce P. Unfortunately, there is no simple way to determine how structural constraints, realized as the removal of links from the network, will affect P. As links are removed, the number of paths that exist between two nodes will depend on which nodes are chosen, and this greatly complicates matters. All we can do at this point is to postulate sample networks and examine their behavior. One such network already has been proposed, namely the one depicted in Figure 7.1.

If the proposer is allowed to search through the network of Figure 7.1 with the only constraint being a limit of six on the path length, the proposer will find only 54 paths between the same 15 pairs of endpoints used above—a decrease from the 5,941,140 paths found in a fully-connected network with the same number of nodes. Finally, applying the constraints under which the proposer normally runs (i.e., causal or intentional relations between events, no paths between competing word senses, etc.) as in the example of Chapter 5, the proposer finds only 33 paths between nine pairs of nodes—12 orders of magnitude smaller than the worst-case value for P given the same number of nodes.

This is, of course, only one case; we could generate examples in which the numbers are worse. As networks are scaled up, the problem associated with uncontrolled spreading activation or marker-passing search becomes increasingly ugly. Nevertheless, it appears that the proposer's search problem is manageable if reasonable assumptions are made about the organization of memory and the depth of search.

7.3 Searching for the best interpretation

With all constraints in effect, the proposer finds 33 paths between nine pairs of nodes when processing Text 5. This does not mean that the search space generated by the proposer consists of only 33 solutions, for the possible solutions which the filter will consider are made up of *combinations* of paths. Given the aforementioned restrictions on the proposer and on memory, there are at most $2^{33} = 8,589,934,592$ possible combinations of paths. This large number may be reduced by considering the structure and meaning of memory to a mere 6912 possible interpretations, an improvement of six orders of magnitude.[3] This number is computed in the following manner. Any open class word in the text may activate one or more nodes in the semantic memory. The proposer finds paths joining the nodes activated by one word to the nodes activated by another word. The discovered paths are different and competing relationships between a pair of words, and the filter can choose only one of these paths for inclusion in its representation. The set of paths joining two words are interchangeable and therefore represent an equivalence class of paths. As ATLAST processes more words from the text it establishes more equivalence classes. At any given time during the processing, ATLAST's active interpretation may include at most one representative from each of the equivalence classes established up to that point. After reading all of Text 5, ATLAST has established seven such equivalence classes; these classes are depicted in Table 7.1.

Recall that Table 7.1 shows three paths connecting `EMBASSY` to `SEARCH`, eight connecting either `INSECT` or `MICROPHONE` to `SEARCH`, and so on. Since ATLAST's final interpretation must include exactly one path from each of these classes, the

[3]Many of these interpretations will be nonsensical, but they are still theoretically possible.

total number of interpretations possible is the product of the number of paths in each of the classes. In the case of Text 5, the number of possible interpretations is:

$$3 \times 8 \times 12 \times 1 \times 3 \times 4 \times 2 = 6912$$

The filter's task then is to find the single best combination of paths which explains the input text. Its search space is the set of all possible combinations containing exactly one path from each equivalence class. When the proposer discovers a new path and passes it to the filter, it is in effect saying, "Compare the explanation you have now to the explanation that you would have if it included this path and choose the better one." The evaluation of a path represents an opportunity to move one step through the search space: the filter either can stay with the current solution or replace it with the one containing the path being evaluated.

7.3.1 The filter's search method

The filter's style of search is best characterized as a form of *hill-climbing*. The filter's search for an interpretation is incremental, tied to the presentation of new words from the input. With each additional word, the filter uses heuristics to determine which of the interpretations proposed through the processing of the new word is best. Hill-climbing suffers from a number of problems, the most serious being the tendency to stop at a locally optimal solution. On the other hand, hill-climbing search has the advantage of being computationally inexpensive when compared to other search techniques. This combination of traits makes hill-climbing a useful search technique when modeling human behavior, as human behavior is not necessarily optimal behavior (Langley, Gennari, & Iba, 1987).

It has been suggested that ATLAST's method of finding a solution is also a form of backtracking. While ATLAST may reconsider interpretations it has previously rejected, it maintains no record of what has been considered in the past: ATLAST knows only the current solution state and a set of other likely solutions, some of which have been considered and others which have not. Backtracking, unlike ATLAST, relies on a history of previous choices (Charniak & McDermott, 1985; Nilsson, 1980). ATLAST may return to a solution it has held at some earlier time, but it will be through a series of opportunistic moves in the search space, not by directed backtracking. Thus, the filter's search bears at most only a superficial resemblance to backtracking.

7.3.2 The principle of locality

The expectation that the filter's hill-climbing search will arrive at the best interpretation of a specific text without unduly taxing ATLAST's computational resources is based on the fact that text tends to be *focused.* Although this focus changes over time, the change is gradual. At any given time, the particular piece of text being read has a much higher probability of relating to what was just read or will soon be read than it does of relating to what was read long ago or will be read far in the future. This assumption is reflected in the behavior of the retained paths: those retained paths which are subsequently suppressed after some interval represent interpretations which might have been plausible at some earlier time but have been deemed implausible with the passage of time and more text. The active and retained paths therefore represent a slowly-changing *window of activity* in memory.

The notion of a window of activity is known elsewhere in computer science. Virtual memory operating systems can take advantage of the tendency of a computer program to favor a subset of its instructions at any time during execution; this tendency is known as the *principle of locality* (Denning, 1970). The program is divided into units called pages. Because of the principle of locality, only a small subset of a program's pages need be in memory for the program to be executed. This set of pages is called the *working set*. When an instruction from within the working set refers to a page not in the working set, a page from the working set must be removed and the new page added (assuming a limit on the number of pages in the working set). If the program is well-constructed the working set will change gradually. Conversely, if the program is poorly-constructed and does not adhere to the principle of locality, the working set may change so rapidly that the operating system spends far more time swapping pages than executing the program.

As with the management of retained paths, the management of an operating system's working set requires a heuristic for deciding which of its members should be removed (i.e., which is least likely to be referenced in the future) when the operating system encounters a reference to something outside the working set. A heuristic for removing a page from the working set may be based on the length of time since the page was last referenced, for example, with the least recently used page being removed first. For both the operating system and ATLAST, the heuristic may be fallible. In the operating system, removing the wrong page means extra work must be done to bring that page back into the working set when it is referenced again. In ATLAST, suppressing the wrong path may lead to an incorrect interpretation of text, but a more comprehensive model (or a human reader) might be forced into conscious problem solving or rereading of the text at this point.

SEARCH to EMBASSY	INSECT or MICROPHONE to SEARCH	INSECT or MICROPHONE to EMBASSY	SEE to SEARCH	SEE to SECRETARY	INSECT to SEE	INSECT to SECRETARY
active paths						
path2	path4	path8	path23	path24	path28	path29
retained paths						
path0	path5 path6 path11	path7 path17 path19 path20 path21 path22		path25 path26	path27 path30 path31	path32
suppressed paths						
path1	path3 path9 path10 path12	path13 path14 path15 path16 path18				

Table 7.2: Inference paths discovered during the processing of Text 5, grouped by path status at completion of processing.

7.3.3 Controlling the filter's search

By the time the processing of a text is completed, the paths found by the proposer can be grouped into three categories: the active paths, the retained paths, and the suppressed or inactive paths. In the example of Text 5, there are 7 active paths, 16 retained paths, and 10 suppressed paths; these are shown in Table 7.2.

The search space of solutions or interpretations can be categorized in a similar fashion. There is only one active interpretation at any given time; this is the set of active paths. The set of retained interpretations consists of all interpretations which contain no suppressed paths and at least one retained path. The number of paths in this category can be computed by calculating the number of interpretations represented by all active and retained paths and subtracting one for the combination of all active paths. For Text 5 this number is $2\times4\times7\times1\times3\times4\times2-1 = 1343$. The set of suppressed interpretations consists of any interpretation containing a suppressed path. This value may be computed indirectly by subtracting the number of active and retained interpretations from the number of all possible interpretations; there are 6912 − 1344 = 5568 suppressed interpretations at the end of Text 5.

With these ideas in mind we can now re-examine, from a performance perspective, the significance of either retaining or suppressing a path in ATLAST. Recall that a retained path is re-evaluated by the filter whenever it shares enough nodes with another path being evaluated—a path which has been newly discovered by the proposer or an older path which has been rediscovered. On the other hand, suppressed paths are treated as if they had never been discovered; consequently, the filter is never reminded of them. Because suppressed paths have little probability of being re-evaluated, any interpretation containing a suppressed path similarly has little probability of being considered by the filter.[4] By comparison, the interpretations that can be constructed by exchanging a retained path for an active path, which is another way of describing the set of retained interpretations, should have a much higher probability of being considered. Thus, the active, retained, and suppressed solutions can also be viewed respectively as the interpretation currently considered to be correct, the interpretations likely to be worth considering in the future, and the interpretations determined to be no longer useful.

Referring again to the example of Text 5, the filter suppresses 10 inference paths by the end of processing, representing only 30% of the paths found by the proposer. Yet by suppressing only 30% of the paths, the filter has eliminated 80% of the possible interpretations from consideration. Thus conditional retention provides a useful heuristic technique for significantly reducing ATLAST's search space, *assuming the existence of an effective metric for determining which paths are retained and which*

[4]A suppressed path may later be discovered and evaluated during the processing of later text. This could happen, for example, if the words corresponding to the endpoints of the suppressed path were repeated in the text. Thus while the suppression of a path substantially diminishes its chances for re-evaluation, it does not completely rule out the possibility.

are suppressed. Conditional retention enables ATLAST to arrive at an interpretation more efficiently than systems which hold onto *all* possible explanations (e.g., Granger's ARTHUR, 1980a) while still maintaining the ability to change its interpretation in the face of contradictory information, unlike systems which retain *no* alternate explanations (e.g., Hirst's ABSITY, 1988b).

Conditional retention in ATLAST requires a method to determine which paths are retained and which paths are suppressed—in other words, a useful heuristic for pruning the search space. Imagine, as an extreme case, a heuristic which pruned too much and suppressed some of the paths of the correct interpretation. Such a heuristic would be useless. At the other extreme, a heuristic which pruned too little would improve the chances of finding the correct interpretation at the expense of making the filter perform many more evaluations.

Some potentially plausible heuristics may also cause problems. For example, consider a heuristic which suppressed retained paths according to how long they had been retained, with the oldest retained paths being suppressed first. If ATLAST had been using this heuristic while processing Text 5, path2 would be among the first to be suppressed. Consequently, when faced with the information that the embassy staff really was looking for insects, ATLAST would be unable to fix its interpretation. This heuristic was used in an early version of ATLAST until this problem was discovered. The heuristic ATLAST currently uses is based on the number of consecutive times a retained path is re-evaluated without being activated. In a crude way, this heuristic takes into account both the duration of retention (i.e., the number of times the path is re-evaluated) and its demonstrated plausibility as part of an active explanation (i.e., the retained path that is subsequently activated and then retained again starts with a clean slate).

As ATLAST processes a text, the search space grows very large very quickly. The principle of inference path retention provides a means of dividing the search space into likely solutions and unlikely solutions; this helps guide the filter's search through the rapidly expanding search space. It seems reasonable to predict that the size of ATLAST's search space of active and retained interpretations could be held roughly constant if, after some start-up period, retained paths were suppressed at the same rate at which new paths were retained; in fact, an example of this behavior is seen in Figure 7.2 in which the growth of ATLAST's search space of possible interpretations during the processing of Text 5 is compared to the growth of its set of retained interpretations. At first the growth of the number of retained interpretations parallels the growth of all possible interpretations but it then drops off dramatically.

The current pruning heuristic performs well on Text 5, but it fails to give any advantage in the processing of some other texts. For example, ATLAST finds the correct interpretation for Text 10 in Appendix A, but the filter does not suppress any paths along the way. No solutions are pruned from the search space, so there is no advantage over an approach which saves all possible solutions, at least in this case. If the heuristic is modified so that it tolerates fewer re-evaluations of a retained path before it suppresses the path, the filter will suppress some paths but they will be the wrong ones and ATLAST will not recover from its original misinterpretation of Text 10.

Obviously, the heuristic currently used for path suppression is not perfect. Of several different heuristics tested, the one described above seems to be the best, but its performance is far from convincing. This is not, however, an argument against conditional retention. ATLAST's implementation has followed a minimalist approach, beginning with a simple and stupid model, adding "intelligence" to it as needed but

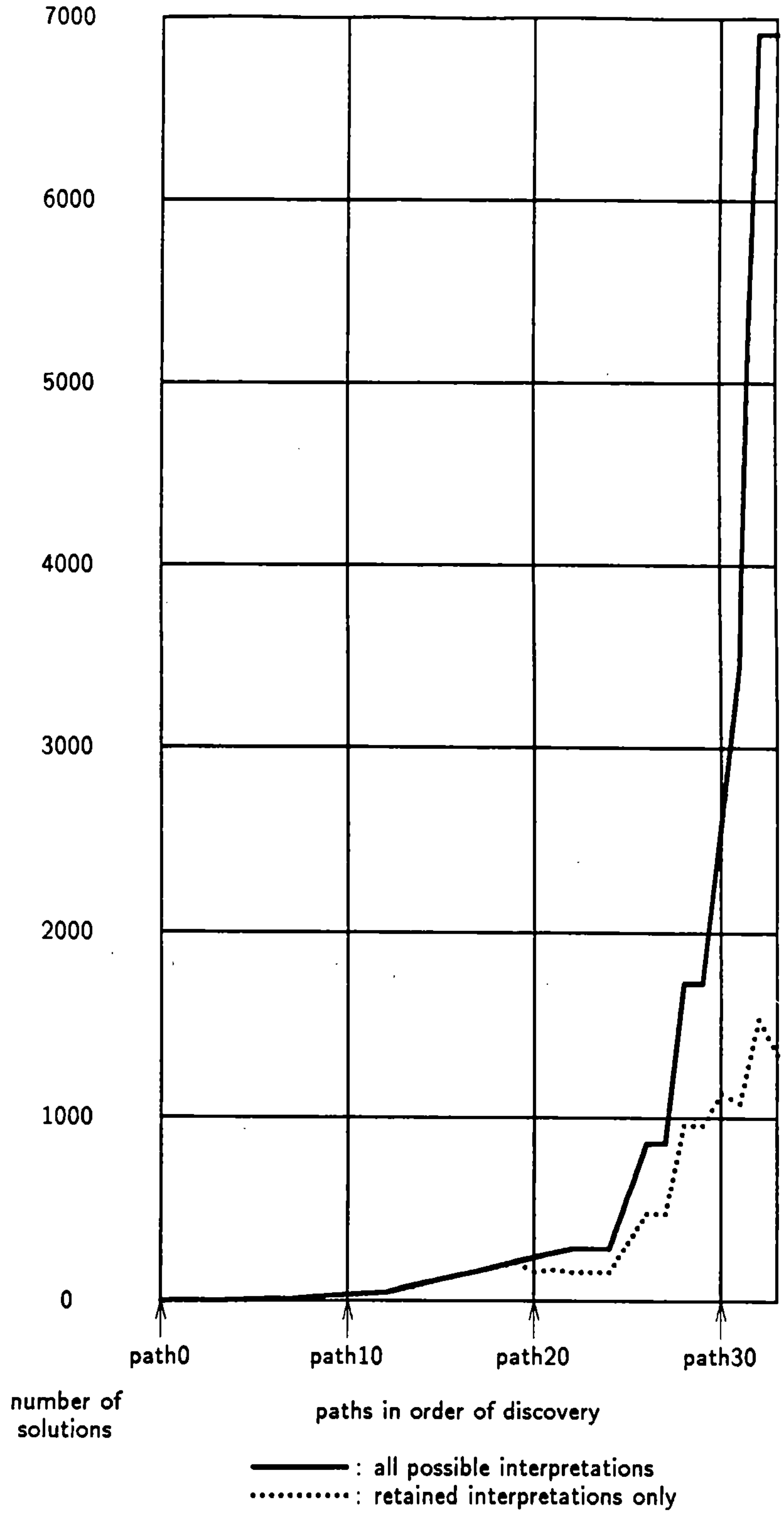

Figure 7.2: The growth of the space of possible interpretations during the processing of Text 5.

staying within the realm of psychological validity at the same time. The goal in building ATLAST has always been to test the plausibility of ideas about inference decisions and error recovery, not to construct a comprehensive language processing system. Therefore, in the course of building ATLAST, many simplifying assumptions have been made about the representation, the processes, and the interactions between them. It is possible that the repercussions of these assumptions are being felt here: the suppression heuristic could probably take advantage of both knowledge and processes which do not exist in this model, suggesting not the invalidity of the conditional retention theory but merely the need for more programming.

7.3.4 Resource consumption

The point of any heuristic search is to maximize the probability of achieving some goal while at the same time attempting to minimize the consumption of one or more resources. These resources are typically defined in terms of computational effort and memory usage. In the filter's search for the best interpretation of a text, computational effort can be measured as the number of path evaluations performed, which is the filter's most computationally-intensive task. Memory usage can be measured as the number of retained inference paths.

In processing Text 5, ATLAST selects one interpretation from the 6912 interpretations that are possible. To accomplish this task, ATLAST does not evaluate every interpretation individually; instead it is able to eliminate potentially large groups of interpretations each time it decides not to activate a path. On the other hand, paths are often evaluated more than once, because the decision to retain a path is only a temporary rejection of a set of interpretations. The number of evaluations performed is directly related to two variables: (1) the maximum duration of path retention and

(2) the minimum degree of similarity required between two paths for the evaluation of one to force the evaluation of the other. As implemented in ATLAST, these variables are represented as two now familiar parameters: (1) the maximum number of consecutive unsuccessful evaluations allowed for a retained path before it is suppressed and (2) the minimum number of nodes shared between a path being evaluated and another retained path. This relationship is depicted graphically in Figure 7.3. The data points are generated by running ATLAST 147 times on Text 5, each time with a different pair of parameter settings.

As the duration of path retention is increased, the number of retained paths increases as well. A greater number of retained paths means a greater number of paths subject to re-evaluation via shared nodes with other paths being evaluated. This in turn results in an increase in the total number of path evaluations, which appears as the surface's positive slope along the y-axis in Figure 7.3. On the other hand, an increase in the number of shared nodes required to indirectly trigger re-evaluation of a retained path makes it more difficult to establish sufficient similarity between any two paths. This results in fewer path evaluations, and is reflected in the negative slope of the surface along the x-axis in the figure.

Most of the cases shown in Figure 7.3 do not result in a correct interpretation.[5] Of the 147 cases shown, only 40 resulted in a correct interpretation. These cases are isolated in Figure 7.4. In this figure, the intersections above the "floor" indicate the parameter settings giving a correct interpretation.

[5] A correct interpretation is the set of all nodes in ATLAST's memory, and the links which join them, that conveys the meaning intended by the author of the text. The correct interpretation must account for all actors, actions, and objects that are explicitly represented in the text, as well as those which can be easily inferred, but must not include other nodes. The correct interpretation must not include contradictions or unresolved ambiguities, and it must be a single interconnected subnetwork.

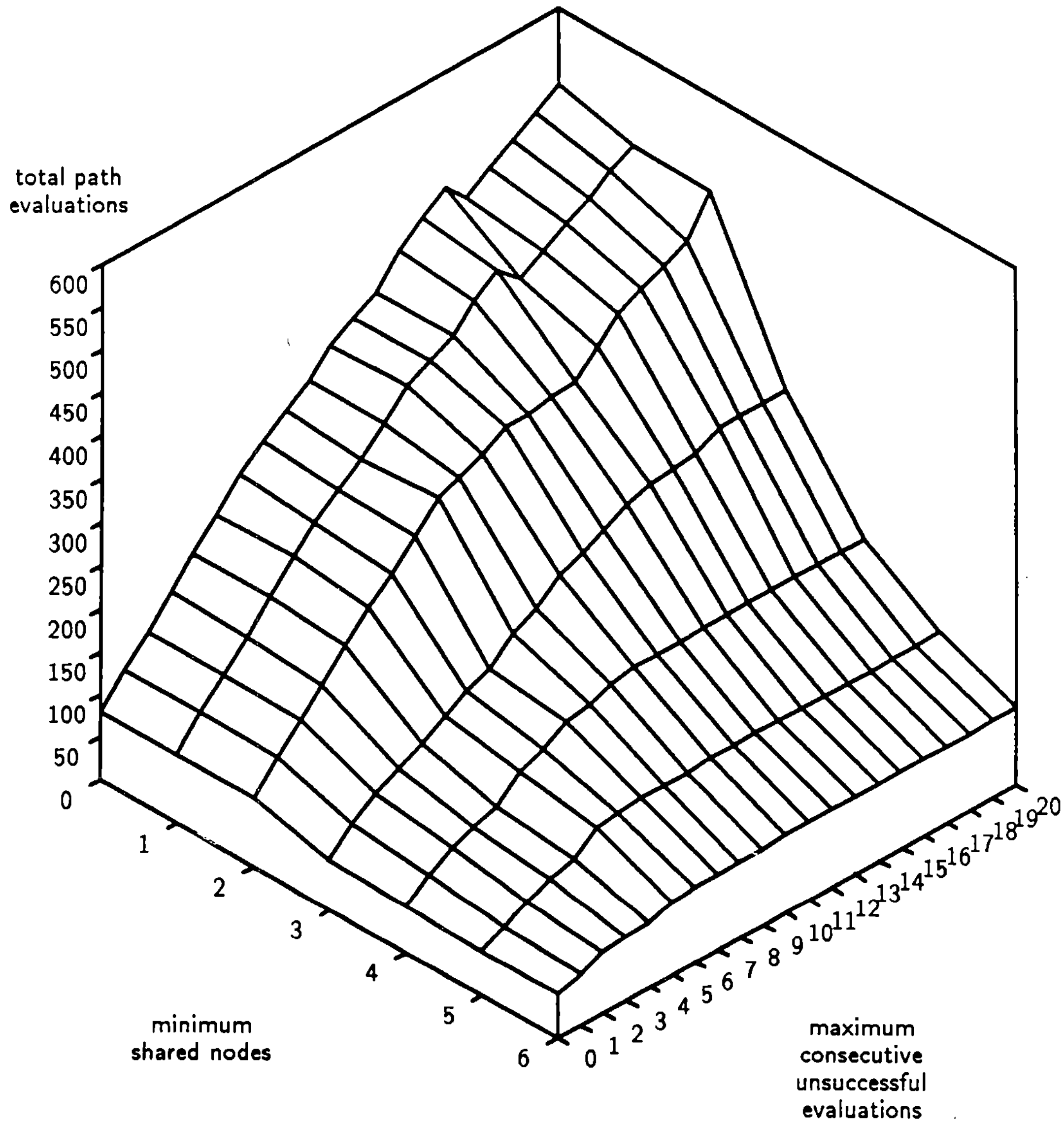

Figure 7.3: Number of path evaluations performed during the processing of Text 5 as conditional retention parameters are varied: all interpretations.

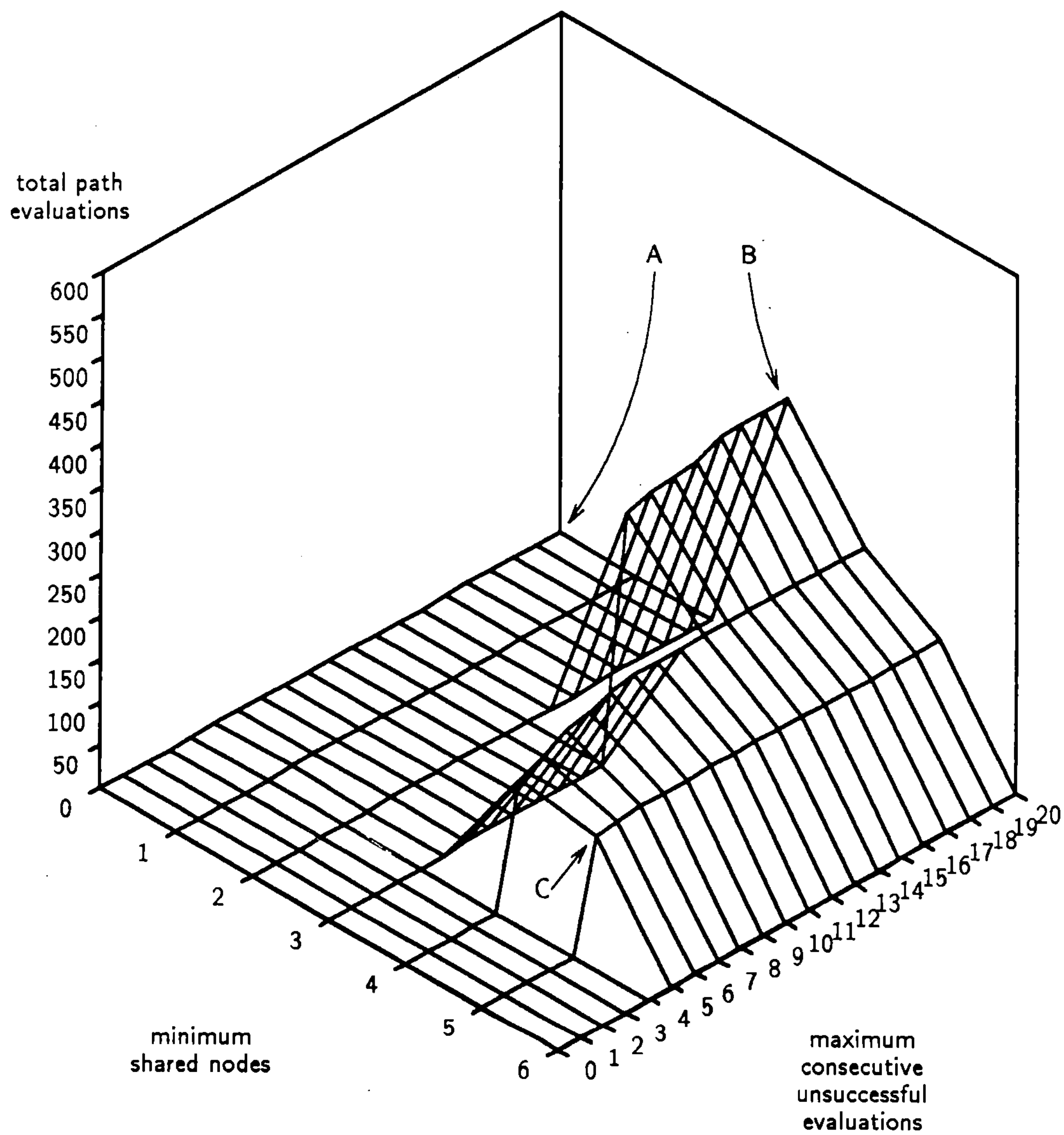

Figure 7.4: Number of path evaluations performed during the processing of Text 5 as conditional retention parameters are varied: correct interpretations only.

One might be surprised by some of the parameter values that did not result in a correct interpretation. For example, with the minimum number of shared nodes set to three and the maximum number of unsuccessful evaluations set to twenty (point B of Figure 7.4), ATLAST finds the correct interpretation of Text 5. Yet when the minimum number of shared nodes is set to zero (point A), so that *every* retained path is re-evaluated, ATLAST is unable to find the correct interpretation. How could ATLAST fail in the latter case when it appears to be doing a more thorough job of reconsidering retained paths? The problem is that decreasing the value of the maximum number of unsuccessful evaluations will not only increase the number of retained paths being re-evaluated on any given cycle, it will also increase the expected frequency with which any given path will be re-evaluated. If those re-evaluations do not result in that path being re-activated, it will be suppressed. At point B, a path necessary for a correct interpretation was still among the retained paths when it was needed. At point A, however, that same path was unavailable because it had been re-evaluated unsuccessfully more than twenty times and was suppressed.

Ideally, ATLAST should arrive at a correct final interpretation using the least number of evaluations possible. In Figure 7.4, that minimum occurs when both the minimum number of shared nodes and the maximum number of unsuccessful evaluations are five (point C). At the same time, however, we want to minimize the number of paths retained by ATLAST during the processing. The number of retained paths is also affected by changes to the two variables; the effects of such changes are shown in Figure 7.5.

Figure 7.5 demonstrates that the number of retained paths not only increases as the duration of path retention is increased, but it also grows as the number of shared nodes is increased. Increasing the minimum degree of similarity required for

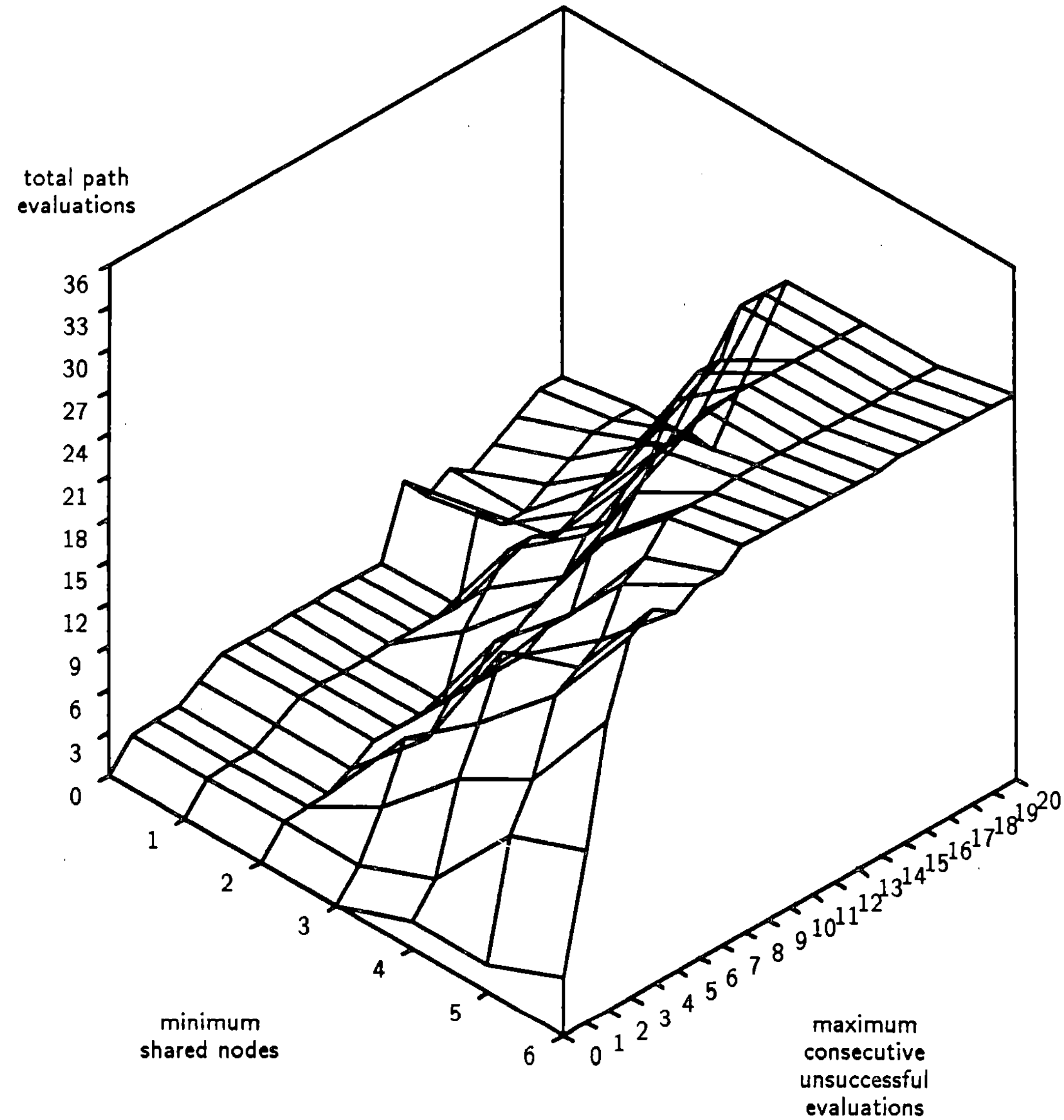

Figure 7.5: Number of paths retained during the processing of Text 5 as conditional retention parameters are varied: all interpretations.

re-evaluation results in fewer path evaluations. As evaluations occur less frequently, the time between a retained path's re-evaluations will grow, and the path will be retained for a longer time period, although the number of consecutive unsuccessful re-evaluations required for suppression of that path will not have changed.

The total number of retained paths is plotted once more in Figure 7.6, but with only the parameter settings which result in a correct interpretation. Again, the desired minimum occurs at the same setting of the parameter for duration of path retention, but at a slightly different value for the number of shared nodes required to trigger path re-evaluation: five and four (point D), respectively, in this case versus five and five in the previous case (point C of Figure 7.4). Thus, there is no single parameter setting which minimizes both the number of path evaluations and the number of retained paths, but it is apparent that the best performance is achieved when the duration of path retention is relatively small.[6] Both ATLAST's computational and memory resources are taxed more heavily as the set of retained paths grows. However, the duration of path retention must not be so short that paths are suppressed almost as soon as they are retained, else ATLAST will not be able to recover from erroneous inference decisions.

7.4 Conclusion

This chapter has viewed ATLAST as a search process whose goal is to find the best interpretation of a text. In fact, ATLAST's search process actually has two

[6]Figure 7.6 also contains a plateau which indicates an upper bound on the number of retained paths. This plateau is merely an artifact of the text, the memory network, and the limits on marker-passing. Regardless of the values of the parameters for duration of path retention and number of shared nodes, the proposer will find only 33 paths for evaluation. Seven of these paths make up the correct interpretation, leaving at most 26 paths that can be retained. The plateau reflects the parameter settings which result in the retention of all 26 paths.

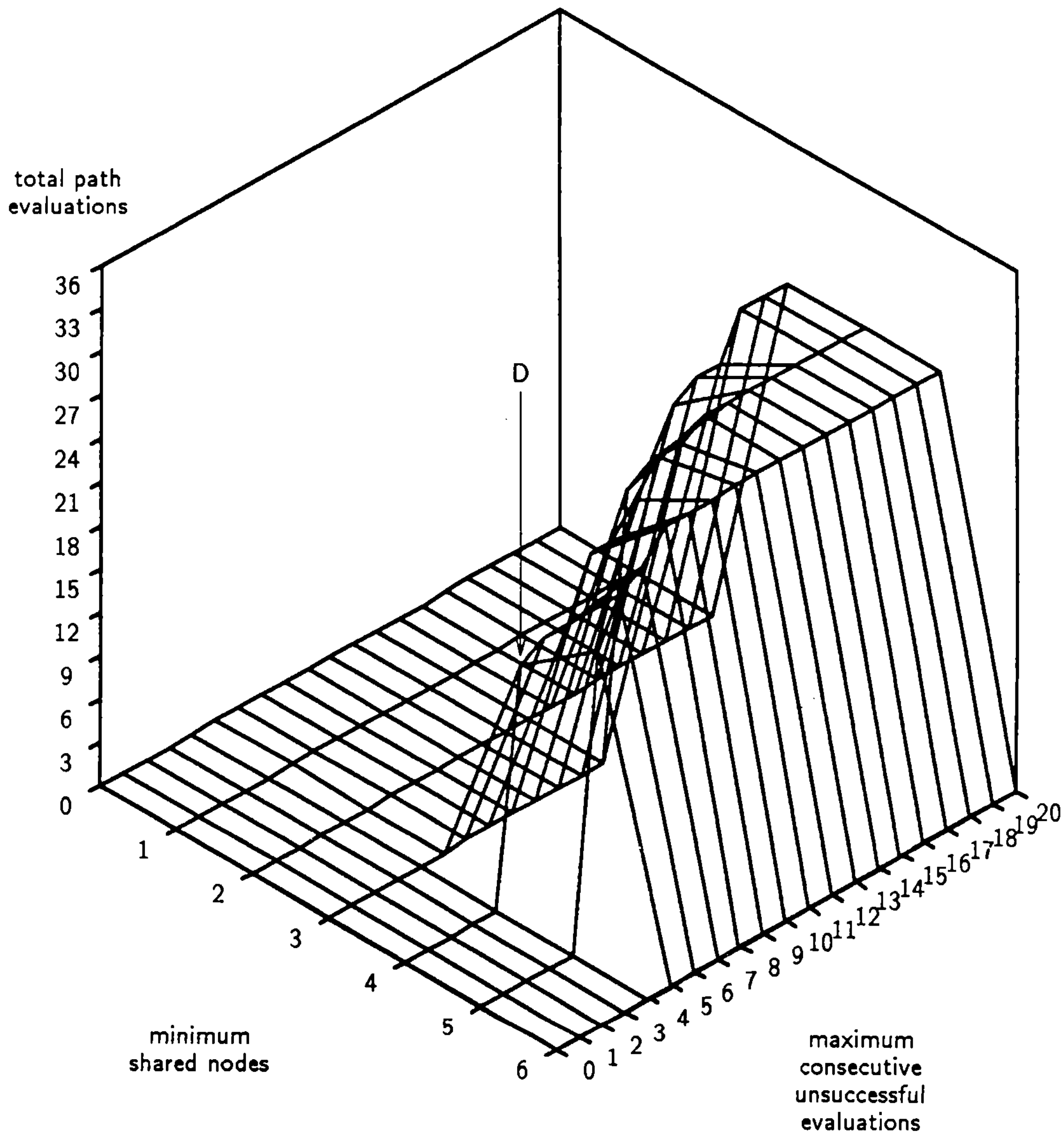

Figure 7.6: Number of paths retained during the processing of Text 5 as conditional retention parameters are varied: correct interpretations only.

components: the proposer's search for inference paths suggested by a text, and the filter's search for a single interpretation of the text made up of a subset of those inference paths.

The proposer uses marker-passing in a breadth-first intersection search for inference paths. There has been some speculation that a marker-passing search in a very large memory network may inundate a path evaluator with a huge number of unimportant paths (Charniak, 1986). This chapter has demonstrated that reasonable assumptions about the organization of memory and the distance that markers may be passed greatly reduce the likelihood of a catastrophic overload of the path evaluator. In other words, the efficiency of marker-passing search is governed not by the size of the network but by the structure of the network and the constraints on marker-passing.[7]

To find the set of paths which best explains a text, the filter uses a hill-climbing search. Hill-climbing can lead to a locally optimal solution, which in many cases is undesirable. Human text understanding, however, seems to benefit if the text follows a principle of locality. Common sense tells us that a passage in which transitions between topics are few and gradual is far more readable than one in which the transitions are numerous and abrupt. Manuals of writing style remind us that, for the sake of readability, a paragraph should have a single topic, and that the beginning of the paragraph should signal the transition from one topic to another. Thus, a search process that finds a locally optimal interpretation may not be adequate for other tasks, but it may be exactly what is needed for text understanding.

[7]Cohen and Kjeldsen (1987) have come to a similar conclusion in their work on an information retrieval system using constrained spreading activation, as has Jones (in preparation) in his work on a spreading activation approach to problem solving.

In summary, ATLAST's component search techniques, marker-passing and hill-climbing, may not be ideally suited to many problems. However, one can make assumptions about the problem of text understanding which minimize the disadvantages of these two search techniques. These assumptions should hold regardless of the size of the memory network. Thus, ATLAST's processing principles should work efficiently in large networks as well as small ones.

Chapter 8

Conclusion

8.1 Summary

Conditional retention is a theory of how a human language understander recovers from an incorrect choice of word meaning. According to this theory, all meanings of an ambiguous word are retrieved and the context-appropriate meaning is chosen. The other meanings are not forgotten but are retained for a certain period of time. If the choice of meaning is contradicted by later text, the retained meanings are reconsidered and a new one is selected. The benefit of conditional retention is that it enables the understander to recover from many inference errors without incurring the cost of backtracking and reprocessing.

Conditional retention is compatible with the prevalent active suppression theory of lexical access, and it answers questions which the active suppression theory cannot. Retention of alternative word meanings after a choice has been made is strongly supported by two experiments (Burgess & Simpson, 1988; Holbrook & Eiselt, in preparation).

ATLAST is a computational model of language understanding which incorporates a theory of recovery from erroneous lexical inference decisions based on the principles of conditional retention. In addition, the ATLAST framework, particularly the mechanism of conditional retention, has been shown to facilitate recovery from erroneous pragmatic inference decisions, thus demonstrating that the proposed mechanisms have both utility and generality. Furthermore, ATLAST demonstrates how individual differences in the conditional retention mechanism might explain individual differences in pragmatic inference decisions.

Although this research offers explanations for a broad range of linguistic and related phenomena, it by no means answers all questions about language understanding. Some of the more important questions raised by ATLAST are discussed in the following sections.

8.2 Open questions

Work on ATLAST has emphasized processing issues over representation issues, but representation issues are by no means unimportant. ATLAST uses a simple relational memory for its representation of semantic knowledge, and its inference evaluation process relies far more on the structure of memory (e.g., the number of links in a path) than on the content of memory (e.g., the meaning of a path) in arriving at an interpretation of a text.

This emphasis on the structure of memory over its content raises concerns about ATLAST's dependence upon a particular representation scheme. For example, ATLAST's preference for shorter inference paths, regardless of the relationship represented by those paths, is essential to finding the correct interpretation of a text. Yet a careless insertion or deletion of a node and link while constructing ATLAST's memory network can have a significant effect on that interpretation. Does this apparent brittleness call into question ATLAST's credibility as a cognitive model? The simple answer is that ATLAST is no more susceptible to accidents in the construction of its memory than is a model using any other approach; it would be just as easy to misrepresent some important attribute in a model which relied more on memory content than on structure. Therefore, ATLAST's reliance on a particular representation scheme makes it no less credible than any other cognitive model.

The answer given above ignores a hidden but more important issue, however: should the processing component of a language understander *not* be dependent upon the representation component? Independence of this kind would certainly ease the fears of those who might worry about memory construction accidents, and a language processing system built under such a constraint would be extremely portable and able to work with any vocabulary available. But these are software engineering issues, not cognitive science issues. From the latter perspective, we must ask if it is reasonable to expect that the human language processing component has not evolved to take the best possible advantage of the implementation specifics of its representational counterpart. In other words, will just any representation scheme suffice, either in the human brain or in cognitive models? The answer to both questions is no, and this answer is reflected in the work of many NLU researchers: the processing components of some earlier language understanding models took great advantage of shorter path effects that emerged from specific representation schemes (Quillian, 1969; Wilks, 1978), and in some more recent models the representation and processing components are one and the same (Cottrell, 1985; Pollack, 1987).

How then is semantic knowledge represented in the human language understander? ATLAST might be able to shed some light on this question in the future.

ATLAST currently processes simple examples using a semantic memory in which lexical and pragmatic knowledge are uniformly represented, and in which the structure of memory is as important to understanding as is the content of memory. More complicated examples, however, will undoubtedly require a greater reliance on the content of memory and the principles of its organization (e.g., scripts, themes, etc.). Robust relational memory schemes incorporating these and other factors could be constructed and tested with ATLAST. The more plausible models of human memory would be those that allowed ATLAST to interpret texts of greater difficulty, maintain adherence to the four constraints on understanding described in Chapter 2, and explain additional linguistic phenomena. These studies might in turn lead to predictions which could be tested through experiments with human subjects, eventually leading to new insights into human memory.

Another open question has to do with the functional independence of syntactic and semantic processing. The texts that ATLAST can understand are, from a syntactic viewpoint, uninteresting. ATLAST parses only simple sentences in subject-verb-object order; it cannot handle interrogatives, imperatives, negatives, or passives, for example. These deficits could be corrected through the addition of a great deal of syntactic knowledge, but there is perhaps more to be learned from an investigation into the relationship between the syntactic and semantic processors. This dissertation has taken the position that these processors can function independently in unusual circumstances, but this position dodges a most important question: to what extent do the processors interact under more normal conditions? As it is currently implemented, ATLAST permits little interaction between syntax and semantics, but it has been constructed specifically so as not to preclude a greater degree of interaction. Thus, ATLAST can be used as a framework for studying various methods of interaction between the syntactic and semantic processors. The results of these studies can then be compared to human performance on the same tasks and possibly shed new light on the issue of functional independence.

8.3 The connectionist connection

Another issue that might be viewed as an open question is that of the implementation of ATLAST. ATLAST was built as a rule-based symbolic processing model with marker-passing, despite the growing popularity of connectionist or parallel distributed processing models. In fact, one anonymous reviewer of an earlier paper on conditional retention offered the opinion that conditional retention "cries out" for a connectionist explanation.

Undoubtedly, there would be some benefits to be gained by a connectionist implementation of ATLAST. First, ATLAST would be better received by those who believe that the only good model is a connectionist model. Second, ATLAST would lose the brittleness that results from its sensitivity to the ordering of the inference evaluation metrics. Because ATLAST's inference evaluation mechanism is a serial process, it must be told in advance the order in which it should apply the metrics to a competing pair of inference paths. A connectionist implementation, in which the functional equivalents of ATLAST's evaluation metrics are built into the architecture, would eliminate any dependence on an arbitrary ordering of rules. A third benefit would be the reduction of ATLAST's computational overhead: there would no longer be a need for the centralized tracking of retained paths.

Recall, though, from the discussion in Chapter 1, that the theory described in this dissertation was intended to correspond to Marr's (1982) computational theory level, and that the ATLAST program was intended to correspond to Marr's representation and algorithm level. It was never the intent of this dissertation to address issues at the implementation level. However, the benefits described above are to be gained primarily at that same level.[1] While these implementation level issues may well be worth pursuing, connectionist implementations pose a unique set of problems for the natural language researcher (Charniak, 1987). Time and effort spent on solving those problems would only detract from work on the higher level issues that ATLAST was intended to address. Consequently, a less problematic implementation has been employed in ATLAST, and it has served well.

8.4 The end

Solving the mysteries of human language understanding inevitably requires an answer to the question of how the language understander resolves ambiguity, for human language is indisputably ambiguous. But ambiguity leads to choices between possible explanations, and choice opens the door for mistakes. Unless we are willing to believe that the human language understander makes the right choice the first time and every time, any explanation of ambiguity resolution must be considered incomplete if it does not also account for recovery from an incorrect decision.

Many models of natural language understanding have dealt with lexical ambiguity resolution in some form, but ATLAST is one of the few to have addressed the associated problem of error recovery. ATLAST's ability to recover from an erroneous lexical decision stems from its ability to retain the word meanings not chosen for a

[1]The benefit of greater acceptance by the connectionist contingent is gained at what might best be described as the *political* level, a level which Marr did not describe but is nevertheless useful in understanding computational models of cognitive processes.

short period after it selects the apparently context-appropriate meaning of an ambiguous word. The short-term retention of possible lexical inferences permits ATLAST to recover from incorrect decisions without backtracking and reprocessing text, and without keeping a record of possible choices indefinitely.

The principle of conditional retention provides a solution to the problem of error recovery which is compatible with current psycholinguistic theories of lexical disambiguation. Furthermore, the existence of some form of retention in human lexical processing is supported by the results of experiments with human subjects. If we assume a uniform representation for all semantic knowledge, the theory of conditional retention can be extended to offer an explanation of recovery from erroneous pragmatic inference decisions as well.

The ATLAST model has served as a platform for refining the conditional retention theory, demonstrating its plausibility, and exploring its implications. In addition, ATLAST illustrates the importance that assumptions play in building models of cognitive processes. The constraints one chooses to adopt or ignore in building a process model directly influence that model's architecture, its behavior, and ultimately its usefulness as a theory of human cognition. Early models of natural language understanding tend to address related phenomena at a single level of the understanding process. As the models begin to address a more diverse set of linguistic phenomena and cut across the perceived levels of processing, common themes emerge. We see memory organized in relational networks, spreading activation search mechanisms, and different processing components running concurrently and, when necessary, independently. Certainly this approach to modeling human language understanding may eventually prove to be wrong, but for the time being it seems to provide better coverage of existing data than other approaches. The relative ease with which this framework can be expanded to account for recovery from erroneous inferences, as demonstrated with ATLAST, further reinforces the belief that this framework represents an important step in the right direction.

References

Abelson, R.P., and Reich, C.M. (1969). Implicational molecules: A method for extracting meaning from input sentences. *Proceedings of the International Joint Conference on Artificial Intelligence*, 641-647.

Anderson, J.A., & Hinton, G.E. (1981). Models of information processing in the brain. In G.E. Hinton & J.A. Anderson (Eds.), *Parallel models of associative memory* (pp. 9-48). Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J.R. (1976). *Language, memory, and thought.* Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J.R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Birnbaum, L. (1985). Lexical ambiguity as a touchstone for theories of language analysis. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 815-820.

Birnbaum, L., & Selfridge, M. (1981). Conceptual analysis of natural language. In R.C. Schank & C.K. Riesbeck (Eds.), *Inside computer understanding: Five programs plus miniatures* (pp. 318-353). Hillsdale, NJ: Lawrence Erlbaum.

Burgess, C., & Simpson, G.B. (1988). Neuropsychology of lexical ambiguity resolution: The contribution of divided visual field studies. In S.L. Small, G.W. Cottrell, & M.K. Tanenhaus (Eds.), *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence* (pp. 411-430). San Mateo, CA: Morgan Kaufmann.

Caramazza, A., & Berndt, R.S. (1978). Semantic and syntactic processes in aphasia: A review of the literature. *Psychological Bulletin*, *85*, 898-918.

Carpenter, P.A., & Daneman, M. (1981). Lexical retrieval and error recovery in reading: A model based on eye fixations. *Journal of Verbal Learning and Verbal Behavior*, *20*, 137-160.

Carroll, D.W. (1986). *Psychology of language.* Monterey, CA: Brooks/Cole.

Charniak, E. (1978). On the use of framed knowledge in language comprehension. *Artificial Intelligence, 11*, 225-265.

Charniak, E. (1981). A parser with something for everyone. *Proceedings of the Third Annual Conference of the Cognitive Science Society*, 231-233.

Charniak, E. (1983). Passing markers: A theory of contextual influence in language comprehension. *Cognitive Science, 7*, 171-190.

Charniak, E. (1986). A neat theory of marker passing. *Proceedings of the Fifth National Conference on Artificial Intelligence*, 584-588.

Charniak, E. (1987). Connectionism and explanation. *TINLAP-3: Theoretical Issues in Natural Language Processing 3*, 71-74.

Charniak, E., & Goldman, R. (1988). A logic for semantic interpretation. *26th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, 87-94.

Charniak, E., & McDermott, D. (1985). *Introduction to artificial intelligence.* Reading, MA: Addison-Wesley.

Chomsky, N. (1957). *Syntactic structures.* The Hague: Mouton.

Cohen, P.R., & Kjeldsen, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management, 23*, 255-268.

Collins, A.M., & Loftus, E.F. (1975). A spreading-activation theory of semantic processing. *Psychological Review, 82*, 407-428.

Cottrell, G.W. (1985). *A connectionist approach to word sense disambiguation* (Technical Report No. 154). Rochester, NY: University of Rochester, Computer Science Department.

Cottrell, G.W., & Small, S.L. (1983). A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory, 6*, 89-120.

Crain, S., & Steedman, M. (1985). On not being led up the garden path: The use of context by the psychological syntax processor. In D.R. Dowty, L. Karttunen, & A.M. Zwicky (Eds.), *Natural language parsing: Psychological, computational, and theoretical perspectives* (pp. 320-358). Cambridge, England: Cambridge University Press.

Cullingford, R.E. (1978). *Script application: Computer understanding of newspaper stories* (Research Report No. 116). New Haven, CT: Yale University, Department of Computer Science.

Daneman, M., & Carpenter, P.A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, *19*, 450-466.

Dyer, M.G. (1983). *In-depth understanding: A computer model of integrated processing for narrative comprehension.* Cambridge, MA: MIT Press.

Denning, P.J. (1970). Virtual memory. *Computing Surveys*, *2*, 153-189.

Eiselt, K.P., & Granger, R.H. (1987). A time-dependent distributed processing model of strategy-driven inference behavior. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, 704-714.

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

Fahlman, S.E., Hinton, G.E., & Sejnowski, T.J. (1983). Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines. *Proceedings of the National Conference on Artificial Intelligence*, 109-113.

Feldman, J.A. (1981). A connectionist model of visual memory. In G.E. Hinton & J.A. Anderson (Eds.), *Parallel models of associative memory* (pp. 49-81). Hillsdale, NJ: Lawrence Erlbaum.

Feldman, J.A., & Ballard, D.H. (1982). Connectionist models and their properties. *Cognitive Science*, *6*, 205-254.

Forster, K.I. (1976). Accessing the mental lexicon. In R.J. Wales & E. Walker (Eds.), *New approaches to language mechanisms* (pp. 257-287). Amsterdam: North-Holland.

Gardner, H. (1985). *The mind's new science.* New York: Basic Books.

Garfield, J.L. (Ed.). (1987). *Modularity in knowledge representation and natural-language understanding.* Cambridge, MA: The MIT Press.

Gentner, D. (1981). Some interesting differences between verbs and nouns. *Cognition and Brain Theory, 4*, 161-178.

Gigley, H.M. (1983). HOPE—AI and the dynamic process of language behavior. *Cognition and Brain Theory, 6*, 39-88.

Granger, R.H., Jr. (1980a). *Adaptive understanding: Correcting erroneous inferences* (Research Report No. 171). New Haven, CT: Yale University, Department of Computer Science.

Granger, R.H., Jr. (1980b). When expectation fails: Towards a self-correcting inference system. *Proceedings of the First Annual National Conference on Artificial Intelligence*, 301-305.

Granger, R.H., Eiselt, K.P., & Holbrook, J.K. (1983). STRATEGIST: A program that models strategy-driven and content-driven inference behavior. *Proceedings of the National Conference on Artificial Intelligence*, 139-147.

Granger, R.H., & Holbrook, J.K. (1983). Perseverers, recencies and deferrers: New experimental evidence for multiple inference strategies in understanding. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society.*

Granger, R.H., Holbrook, J.K., & Eiselt, K.P. (1984). Interaction effects between word-level and text-level inferences: On-line processing of ambiguous words in context. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, 172-178.

Hall, R.P., & Kibler, D.F. (1985). Differing methodological perspectives in artificial intelligence research. *The AI Magazine, 6*(3), 166-178.

Hendler, J.A. (1986). *Integrating marker-passing and problem-solving: A spreading-activation approach to improved choice in planning* (Technical Report No. CS-86-01). Providence, RI: Brown University, Department of Computer Science.

Hillis, W.D. (1985). *The connection machine.* Cambridge, MA: MIT Press.

Hirst, G. (1984). Jumping to conclusions: Psychological reality and unreality in a word disambiguation program. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, 179-182.

Hirst, G. (1988a). Resolving lexical ambiguity computationally with spreading activation and Polaroid words. In S.L. Small, G.W. Cottrell, & M.K. Tanenhaus (Eds.), *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence* (pp. 73-107). San Mateo, CA: Morgan Kaufmann.

Hirst, G. (1988b). Semantic interpretation and ambiguity. *Artificial Intelligence, 34*, 131-177.

Hobbs, J.R. (1987). World knowledge and word meaning. *TINLAP-3: Theoretical Issues in Natural Language Processing 3*, 20-27.

Hogaboam, T.W., & Perfetti, C.A. (1975). Lexical ambiguity and sentence comprehension. *Journal of Verbal Learning and Verbal Behavior, 14*, 265-274.

Holbrook, J.K. (in preparation). *Studies of inference retention in lexical ambiguity resolution.* Irvine: University of California, Department of Social Sciences.

Holbrook, J.K., and Eiselt, K.P. (in preparation). *Evidence for the retention of unselected meanings of ambiguous words during text processing.* Irvine: University of California, Department of Social Sciences.

Israel, D. (1987). On formal versus commonsense semantics. *TINLAP-3: Theoretical Issues in Natural Language Processing 3*, 134-139.

Jones, R. (in preparation). *A model of retrieval in problem solving.* Irvine: University of California, Department of Information and Computer Science.

Just, M.A., & Carpenter, P.A. (1978). Inference processes during reading: Reflections from eye fixations. In J.W. Senders, D.F. Fisher, & R.A. Monty (Eds.), *Eye movements and the higher psychological functions* (pp. 157-174). Hillsdale, NJ: Lawrence Erlbaum.

Just, M.A., & Carpenter, P.A. (1980). A theory of reading: From eye fixations to comprehension. *Psychological Review, 87*, 329-354.

Kay, P. (1983). *Three properties of the ideal reader* (Berkeley Cognitive Science Report No. 7). Berkeley: University of California, Institute of Cognitive Studies, Cognitive Science Program.

Kegl, J. (1987). The boundary between word knowledge and world knowledge. *TINLAP-3: Theoretical Issues in Natural Language Processing 3*, 28-33.

Kintsch, W. (1974). *The representation of meaning in memory.* Hillsdale, NJ: Lawrence Erlbaum.

Kintsch, W., & van Dijk, T.A. (1978). Toward a model of text comprehension and production. *Psychological Review, 85*, 363-394.

Langley, P., Gennari, J.H., & Iba, W. (1987). Hill-climbing theories of learning. *Proceedings of the Fourth International Workshop on Machine Learning*, 312-323.

Lashley, K.S. (1951). The problem of serial order in behavior. In L.A. Jeffress (Ed.), *Cerebral mechanisms in behavior: The Hixon symposium* (pp. 112-136). New York: Hafner.

Lebowitz, M. (1980). *Generalization and memory in an integrated understanding system* (Research Report No. 186). New Haven, CT: Yale University, Department of Computer Science.

Lucas, M. (1983). Lexical access during sentence comprehension: Frequency and context effects. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society.*

Lytinen, S.L. (1984). *The organization of knowledge in a multi-lingual, integrated parser* (Research Report No. YALEU/CSD/RR 340). New Haven, CT: Yale University, Department of Computer Science.

Lytinen, S.L. (1986). Dynamically combining syntax and semantics in natural language processing. *Proceedings AAAI-86 Fifth National Conference on Artificial Intelligence*, 574-578.

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

Marcus, M.P. (1984). Some inadequate theories of human language processing. In T.G. Bever, J.M. Carroll, & L.A. Miller (Eds.), *Talking minds: The study of language in cognitive science* (pp. 253-278). Cambridge, MA: MIT Press.

Marr, D. (1982). *Vision.* New York: W.H. Freeman.

McClelland, J.L., & Rumelhart, D.E. (1981). An interactive activation model of context effects in letter perception. Part 1, an account of basic findings. *Psychological Review, 88*, 375-407.

McDermott, D.V. (1974). *Assimilation of new information by a natural language-understanding system* (Technical Report No. AI TR-291). Cambridge: Massachusetts Institute of Technology, Artificial Intelligence Laboratory

Meyer, D.E., & Schvaneveldt, R.W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology, 90*, 227-234.

Morgan, C.L. (1977). An introduction to comparative psychology. In Robinson, D.N. (Ed.), *Significant contributions to the history of psychology 1750-1920, series D, comparative psychology, volume II, C.L. Morgan.* Washington, DC: University Publications of America. (Original work published in 1894).

Morton, J. (1969). Interaction of information in word recognition. *Psychological Review, 76*, 165-178.

Morton, J. (1979) Word recognition. In J. Morton & J.C. Marshall (Eds.), *Psycholinguistics 2: Structure and process* (pp. 107-156). London: Paul Elek.

Moyne, J.A., & Kaniklidis, C. (1981). Models of language comprehension. *Cognition and Brain Theory, 4*, 265-284.

Nilsson, N.J. (1980). *Principles of artificial intelligence.* Palo Alto, CA: Tioga Publishing.

Norvig, P. (1983). Six problems for story understanders. *Proceedings of the National Conference on Artificial Intelligence*, 284-287.

Norvig, P. (1987). *Unified theory of inference for text understanding* (Technical Report No. UCB/CSD 87/339). Berkeley: University of California, Computer Science Division (EECS).

Onifer, W., & Swinney, D.A. (1981). Accessing lexical ambiguities during sentence comprehension: Effects of frequency of meaning and contextual bias. *Memory and Cognition, 9*, 225-236.

Orejel-Opisso, J.L. (1984). *Story understanding with WATSON, a computer program modeling natural language inferences using nonmonotonic dependencies* (Technical Report No. T146). Urbana: University of Illinois, Coordinated Science Laboratory, Advanced Automation Group.

O'Rorke, P. (1983). Reasons for beliefs in understanding: Applications of nonmonotonic dependencies to story processing. *Proceedings of the National Conference on Artificial Intelligence*, 306-309.

Phillips, B., & Hendler, J. (1981). Controlling parsing by passing messages. *Proceedings of the Third Annual Conference of the Cognitive Science Society*, 228-230.

Pollack, J.B. (1987). *On connectionist models of natural language processing* (Technical Report No. MCCS-87-100). Las Cruces: New Mexico State University, Computing Research Laboratory.

Pollack, J., & Waltz, D. (1982). Natural language processing using spreading activation and lateral inhibition. *Proceedings of the Fourth Annual Conference of the Cognitive Science Society*, 50-53.

Pylyshyn, Z.W. (1980). Computation and cognition: Issues in the foundations of cognitive science. *The Behavioral and Brain Sciences*, *3*, 111-169.

Quillian, M.R. (1968). Semantic memory. In M. Minsky (Ed.), *Semantic information processing* (pp. 227-270). Cambridge, MA: MIT Press.

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

Rieger, C., Trigg, R., & Bane, B. (1981). ZMOB: A new computing engine for AI. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 955-960.

Riesbeck, C.K. (1975). Conceptual analysis. In R.C. Schank (Ed.), *Conceptual information processing* (pp. 83-156). Amsterdam: North-Holland.

Riesbeck, C.K., & Martin, C.E. (1986). Direct memory access parsing. In J.L. Kolodner & C.K. Riesbeck (Eds.), *Experience, memory, and reasoning* (pp. 209-226). Hillsdale, NJ: Lawrence Erlbaum.

Rumelhart, D.E. (1981). *Understanding understanding* (Technical Report No. 100). San Diego: University of California, Center for Human Information Processing.

Rumelhart, D.E., & McClelland, J.L. (1982). An interactive activation model of context effects in letter perception. Part 2, the contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, *89*, 60-94.

Rumelhart, D.E., & Norman, D.A. (1975). The active structural network. In D.A. Norman & D.E. Rumelhart (Eds.), *Explorations in cognition* (pp. 35-64). San Francisco: W.H. Freeman.

Schank, R.C. (Ed.). (1975). *Conceptual information processing.* Amsterdam: North-Holland.

Schank, R.C., & Abelson, R.P. (1977). *Scripts, plans, goals, and understanding.* Hillsdale, NJ: Lawrence Erlbaum.

Schank, R.C., Collins, G.C., Davis, E., Johnson, P.N., Lytinen, S., & Reiser, B.J. (1982). What's the point? *Cognitive Science*, *6*, 255-275.

Schank, R.C., & Riesbeck, C.K. (Eds.). (1981). *Inside computer understanding: Five programs plus miniatures.* Hillsdale, NJ: Lawrence Erlbaum.

Seidenberg, M.S., Tanenhaus, M.K., Leiman, J.M., & Bienkowski, M. (1982). Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. *Cognitive Psychology, 14*, 489-537.

Seifert, C.M., Robertson, S.P., & Black, J.B. (1982). *On-line processing of pragmatic inferences* (Technical Report No. 15). New Haven, CT: Yale University, Cognitive Science Program.

Shebilske, W.L., & Fisher, D.F. (1983). Eye movements and context effects during reading of extended discourse. In K. Rayner (Ed.), *Eye movements in reading: Perceptual and language processes.* New York: Academic Press.

Simpson, G. (1981). Meaning dominance and semantic context in the processing of lexical ambiguity. *Journal of Verbal Learning and Verbal Behavior, 20*, 120-136.

Simpson, G.B. (1984). Lexical ambiguity and its role in models of word recognition. *Psychological Bulletin, 96*, 316-340.

Simpson, G.B., & Burgess, C. (1985). Activation and selection processes in the recognition of ambiguous words. *Journal of Experimental Psychology: Human Perception and Performance, 11*, 28-39.

Small, S., & Rieger, C. (1982). Parsing and comprehending with word experts (a theory and its realization). In W.G. Lehnert & M.H. Ringle (Eds.), *Strategies for natural language processing* (pp. 89-147). Hillsdale, NJ: Lawrence Erlbaum.

Swinney, D.A. (1979). Lexical access during sentence comprehension: (Re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior, 18*, 645-660.

Tanenhaus, M., Leiman, J., & Seidenberg, M. (1979). Evidence for multiple stages in processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior, 18*, 427-440.

Thibadeau, R., Just, M.A., & Carpenter, P.A. (1982). A model of the time course and content of reading. *Cognitive Science, 6*, 157-203.

Tyler, L.K., & Marslen-Wilson, W.D. (1977). The on-line effects of semantic context on syntactic processing. *Journal of Verbal Learning and Verbal Behavior, 16*, 683-692.

Wagner, A. R. (1981). SOP: A model of automatic memory processing in animal behavior. In N. E. Spear & R. R. Miller (Eds.), *Information processing in animals: Memory mechanisms* (pp. 5-47). Hillsdale, NJ: Lawrence Erlbaum.

Waltz, D.L., & Pollack, J.B. (1984). Phenomenologically plausible parsing. *Proceedings of the National Conference on Artificial Intelligence*, 335-339.

Waltz, D.L., & Pollack, J.B. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, *9*, 51-74.

Wilensky, R. (1978). *Understanding goal-based stories* (Research Report No. 140). New Haven, CT: Yale University, Department of Computer Science.

Wilensky, R. (1983). *Planning and understanding: A computational approach to human reasoning.* Reading, MA: Addison-Wesley.

Wilensky, R. (1984). KODIAK—A knowledge representation language. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, 344-352.

Wilks, Y. (1975a). A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, *6*, 53-74.

Wilks, Y. (1975b). An intelligent analyzer and understander of English. *Communications of the ACM*, *18*, 264-274.

Wilks, Y. (1978). Making preferences more active. *Artificial Intelligence*, *11*, 197-223.

Wilks, Y. (1987). On keeping logic in its place. *TINLAP-3: Theoretical Issues in Natural Language Processing 3*, 124-128.

Winograd, T. (1973). A procedural model of language understanding. In R.C. Schank & K.M. Colby (Eds.), *Computer models of thought and language* (pp. 152-186). San Francisco: W.H. Freeman.

Winograd, T. (1983). *Language as a cognitive process. Volume 1: Syntax.* Reading, MA: Addison-Wesley.

Woods, W.A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, *13*, 591-606.

Appendix A

Pragmatic Inference Processing: Implementation

A.1 Pragmatic disambiguation and recovery

This appendix describes the operation of ATLAST as it performs several tasks in pragmatic inference processing. Using actual output generated by ATLAST during the processing of simplified versions of the sample stories from Chapter 6, this appendix shows how ATLAST makes pragmatic inference decisions, how it recovers from erroneous pragmatic inference decisions, and how it can model either of the two inference processing strategies described previously.

ATLAST resolves pragmatic ambiguity and successfully corrects its mistakes using the same inference evaluation and error recovery mechanism that it uses at the lexical level. These abilities are demonstrated in the processing of a variant of Text 7:

Text 7: John was poor but he owned a gun.
He went to the pawnshop.
He sold the gun.

Because ATLAST does not know about pronouns and understands only the simplest sentences, ATLAST is given this rough approximation:

Text 10: John was poor.
John owned a gun.
John went to the pawnshop.
John sold the gun.

Because of the enormity of the output generated by ATLAST during the processing of Text 10, this example has been edited so that nothing generated by the capsulizer or the proposer appears, and only the more important actions of the filter are present. A diagram of the organization of the semantic memory used by ATLAST in understanding Text 10 is given in Figure A.1.

```
Input text is:

    John was poor.
    John owned a gun.
    John went to the pawnshop.
    John sold the gun.

Ordering of inference evaluation metrics in force:

    MORE-ACTIVATION-METRIC
    SHORTER-PATH-METRIC
    MORE-REINFORCEMENT-METRIC
    MORE-SPECIFIC-METRIC
    NO-DECISION-METRIC

Maximum distance of marker-passing: 3

Distance to pass markers per cycle: 3
Are rejected paths being retained?: t
```

To enable ATLAST to find the correct interpretation of Text 10, two of the parameters have been changed from their values in the examples of Chapter 5. The maximum number of consecutive unsuccessful evaluations that a retained path can endure before it is suppressed has been increased to 17 as anything less results in ATLAST arriving at an inconsistent interpretation. The minimum number of nodes a retained path must share with a path under evaluation so that the retained path

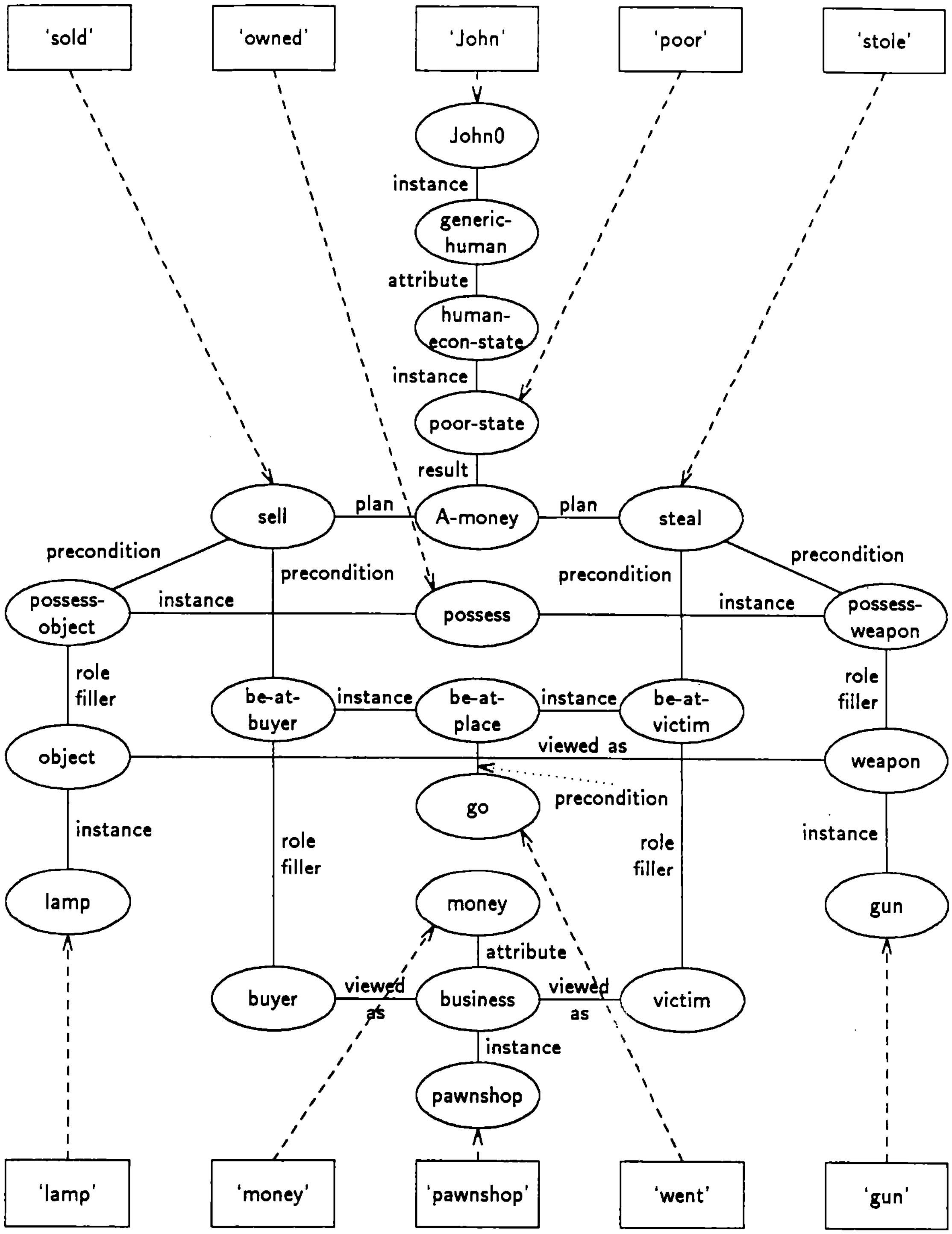

Figure A.1: Memory network for Text 10.

will be re-evaluated has been reduced to 2. Any greater value for this parameter also will result in an inconsistent interpretation. The relationship between the parameter settings and ATLAST's performance is discussed in greater depth in Chapter 7.

```
Max. no. of unsuccessful evaluations: 17
Min. no. of shared nodes to force re-evaluation: 2

Are function words being allocated processing cycles?: nil
Is inference processing forced to complete at periods?: t
```

A.1.1 The first sentence

ATLAST's processing of the first sentence is uneventful. It finds only one path connecting John and poor.

```
  New path discovered: path0
    Path from POOR-STATE to JOHNO
      POOR-STATE is an instance of HUMAN-ECON-STATE
      HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
      GENERIC-HUMAN has the instance JOHNO
      Activating path0

  Active memory structure:

    Paths: (path0)
    Path from POOR-STATE to JOHNO
      POOR-STATE is an instance of HUMAN-ECON-STATE
      HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
      GENERIC-HUMAN has the instance JOHNO

  Pointers to memory structure:

    Event:  event0
    Actor:  (JOHNO)
    Action: (BE)
    Object: (POOR-STATE)
    Direction: nil
```

A.1.2 The second sentence

ATLAST begins working on the second sentence and finds that John possesses something. It then tries to find connections between this fact and the previously processed text. The system infers that John has the goal of obtaining money and that owning something is related to that goal. ATLAST has limited knowledge in this area, however: it knows only that possessing an object of value is a precondition to selling it (path2) and that possessing an object which can be used as a weapon is a precondition to stealing something (path1). ATLAST is unable to decide between the two paths so both are retained.

```
New path discovered: path1
  Path from POSSESS to POOR-STATE
    POSSESS has the instance POSSESS-WEAPON
    POSSESS-WEAPON is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
    Activating path1

New path discovered: path2
  Path from POSSESS to POOR-STATE
    POSSESS has the instance POSSESS-OBJECT
    POSSESS-OBJECT is a precondition of SELL
    SELL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
  No-decision metric -- path1 and path2 are retained
    De-activating path1
    De-activating path2
```

Continuing with the second sentence of Text 10, ATLAST reads that what John owns is a gun. While a gun can be viewed as an object to be sold (path4), ATLAST determines that a more specific use of a gun is as a weapon (path3). The evaluation metrics select path3 over path4, and any retained paths that are sufficiently related to either path3 or path4 are re-evaluated. The path that is consistent with the use of the gun as a weapon (i.e., path1, in which possession of a weapon is a precondition to stealing) is activated. ATLAST has determined that John is going to use his gun to steal money.

```
New path discovered: path3
  Path from GUN to POSSESS
    GUN is an instance of WEAPON
    WEAPON is a role-filler of POSSESS-WEAPON
    POSSESS-WEAPON is an instance of POSSESS
Also reconsidering (path1) due to shared nodes with path3
Also reconsidering (path2) due to tie with path1
    Activating path3
    Activating path1
  No-decision metric -- path1 and path2 are retained
    De-activating path1
    De-activating path2

New path discovered: path4
  Path from GUN to POSSESS
    GUN is an instance of WEAPON
    WEAPON can be viewed as OBJECT
    OBJECT is a role-filler of POSSESS-OBJECT
    POSSESS-OBJECT is an instance of POSSESS
Also reconsidering (path2) due to shared nodes with path4
Also reconsidering (path1) due to tie with path2
  More-specific metric -- path3 more specific than path4
    De-activating path4
    Activating path2
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2
    Activating path1

Active memory structure:

  Paths: (path1 path3 path0)
  Path from POSSESS to POOR-STATE
    POSSESS has the instance POSSESS-WEAPON
    POSSESS-WEAPON is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
  Path from GUN to POSSESS
    GUN is an instance of WEAPON
    WEAPON is a role-filler of POSSESS-WEAPON
```

```
    POSSESS-WEAPON is an instance of POSSESS
  Path from POOR-STATE to JOHNO
    POOR-STATE is an instance of HUMAN-ECON-STATE
    HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance JOHNO

Pointers to memory structure:

  Event:  event0
  Actor:  (JOHNO)
  Action: (BE)
  Object: (POOR-STATE)
  Direction: nil

  Event:  event1
  Actor:  (JOHNO)
  Action: (POSSESS)
  Object: (GUN)
  Direction: nil
```

A.1.3 The third sentence

ATLAST then begins work on the third sentence of the text. After reading `John went` in the context of the previously processed text, ATLAST infers that John is going to some yet undisclosed location to steal money.

```
New path discovered: path5
  Path from GO to POOR-STATE
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-BUYER
    BE-AT-BUYER is a precondition of SELL
    SELL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
Also reconsidering (path2) due to shared nodes with path5
    Activating path5
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2
```

```
New path discovered: path6
  Path from GO to POSSESS
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-BUYER
    BE-AT-BUYER is a precondition of SELL
    SELL has the precondition POSSESS-OBJECT
    POSSESS-OBJECT is an instance of POSSESS
Also reconsidering (path2 path4) due to shared nodes with path6
    Activating path6
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2
  More-reinforcement metric -- path3 has more shared nodes
    than path4
    De-activating path4

New path discovered: path7
  Path from GO to POOR-STATE
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-VICTIM
    BE-AT-VICTIM is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
Also reconsidering (path2) due to shared nodes with path7
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5
    Activating path7
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2

New path discovered: path8
  Path from GO to POSSESS
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-VICTIM
    BE-AT-VICTIM is a precondition of STEAL
    STEAL has the precondition POSSESS-WEAPON
    POSSESS-WEAPON is an instance of POSSESS
Also reconsidering (path5) due to shared nodes with path8
  More-reinforcement metric -- path8 has more shared nodes
    than path6
    De-activating path6
```

```
    Activating path8
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5

Active memory structure:

  Paths: (path8 path7 path1 path3 path0)
  Path from GO to POSSESS
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-VICTIM
    BE-AT-VICTIM is a precondition of STEAL
    STEAL has the precondition POSSESS-WEAPON
    POSSESS-WEAPON is an instance of POSSESS
  Path from GO to POOR-STATE
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-VICTIM
    BE-AT-VICTIM is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
  Path from POSSESS to POOR-STATE
    POSSESS has the instance POSSESS-WEAPON
    POSSESS-WEAPON is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
  Path from GUN to POSSESS
    GUN is an instance of WEAPON
    WEAPON is a role-filler of POSSESS-WEAPON
    POSSESS-WEAPON is an instance of POSSESS
  Path from POOR-STATE to JOHNO
    POOR-STATE is an instance of HUMAN-ECON-STATE
    HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance JOHNO
```

ATLAST completes processing on the remainder of the third sentence and finds that John went to the pawnshop. The system infers that the pawnshop will be the victim in John's plan of stealing money.

```
New path discovered: path9
  Path from PAWNSHOP to GO
    PAWNSHOP is an instance of BUSINESS
    BUSINESS can be viewed as BUYER
```

```
    BUYER is a role-filler of BE-AT-BUYER
    BE-AT-BUYER is an instance of BE-AT-PLACE
    BE-AT-PLACE has the precondition GO
Also reconsidering (path5 path6) due to shared nodes with path9
    Activating path9
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5
  More-reinforcement metric -- path8 has more shared nodes
    than path6
    De-activating path6

New path discovered: path10
  Path from PAWNSHOP to GO
    PAWNSHOP is an instance of BUSINESS
    BUSINESS can be viewed as VICTIM
    VICTIM is a role-filler of BE-AT-VICTIM
    BE-AT-VICTIM is an instance of BE-AT-PLACE
    BE-AT-PLACE has the precondition GO
Also reconsidering (path6 path5) due to shared nodes with path10
  More-reinforcement metric -- path10 has more shared nodes
    than path9
    De-activating path9
    Activating path10
  More-reinforcement metric -- path8 has more shared nodes
    than path6
    De-activating path6
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5

Active memory structure:

  Paths: (path10 path8 path7 path1 path3 path0)
  Path from PAWNSHOP to GO
    PAWNSHOP is an instance of BUSINESS
    BUSINESS can be viewed as VICTIM
    VICTIM is a role-filler of BE-AT-VICTIM
    BE-AT-VICTIM is an instance of BE-AT-PLACE
    BE-AT-PLACE has the precondition GO
  Path from GO to POSSESS
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-VICTIM
```

```
    BE-AT-VICTIM is a precondition of STEAL
    STEAL has the precondition POSSESS-WEAPON
    POSSESS-WEAPON is an instance of POSSESS
  Path from GO to POOR-STATE
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-VICTIM
    BE-AT-VICTIM is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
  Path from POSSESS to POOR-STATE
    POSSESS has the instance POSSESS-WEAPON
    POSSESS-WEAPON is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
  Path from GUN to POSSESS
    GUN is an instance of WEAPON
    WEAPON is a role-filler of POSSESS-WEAPON
    POSSESS-WEAPON is an instance of POSSESS
  Path from POOR-STATE to JOHNO
    POOR-STATE is an instance of HUMAN-ECON-STATE
    HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance JOHNO

Pointers to memory structure:

  Event:  event0
  Actor:  (JOHNO)
  Action: (BE)
  Object: (POOR-STATE)
  Direction: nil

  Event:  event1
  Actor:  (JOHNO)
  Action: (POSSESS)
  Object: (GUN)
  Direction: nil

  Event:  event2
  Actor:  (JOHNO)
  Action: (GO)
  Object: nil
  Direction: (PAWNSHOP)
```

Using its limited knowledge of goals, plans, and preconditions, ATLAST has made the inferences necessary to tie together the pragmatically ambiguous events and states of the first three sentences of Text 10. Up to this point ATLAST has discovered eleven paths, six of which now make up ATLAST's active interpretation while the remaining five are retained. The active interpretation is shown in Figure A.2.

A.1.4 The final sentence

ATLAST now processes the final and contradictory sentence of Text 10. The system had determined previously that John intended to use his gun to steal money from the pawnshop. However, just the first two words of the final sentence, `John sold`, inform ATLAST that it has made the wrong inferences. Initially, ATLAST finds connections between selling and being poor (path11), between selling and possessing something (path12), and between selling and going somewhere (path13). These paths are added to the active interpretation. The inferences are inconsistent with the existing interpretation but are not sufficient to cause ATLAST to revise other inferences.

```
New path discovered: path11
  Path from SELL to POOR-STATE
    SELL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
Also reconsidering (path5 path2) due to shared nodes with path11
    Activating path11
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2

New path discovered: path12
  Path from SELL to POSSESS
```

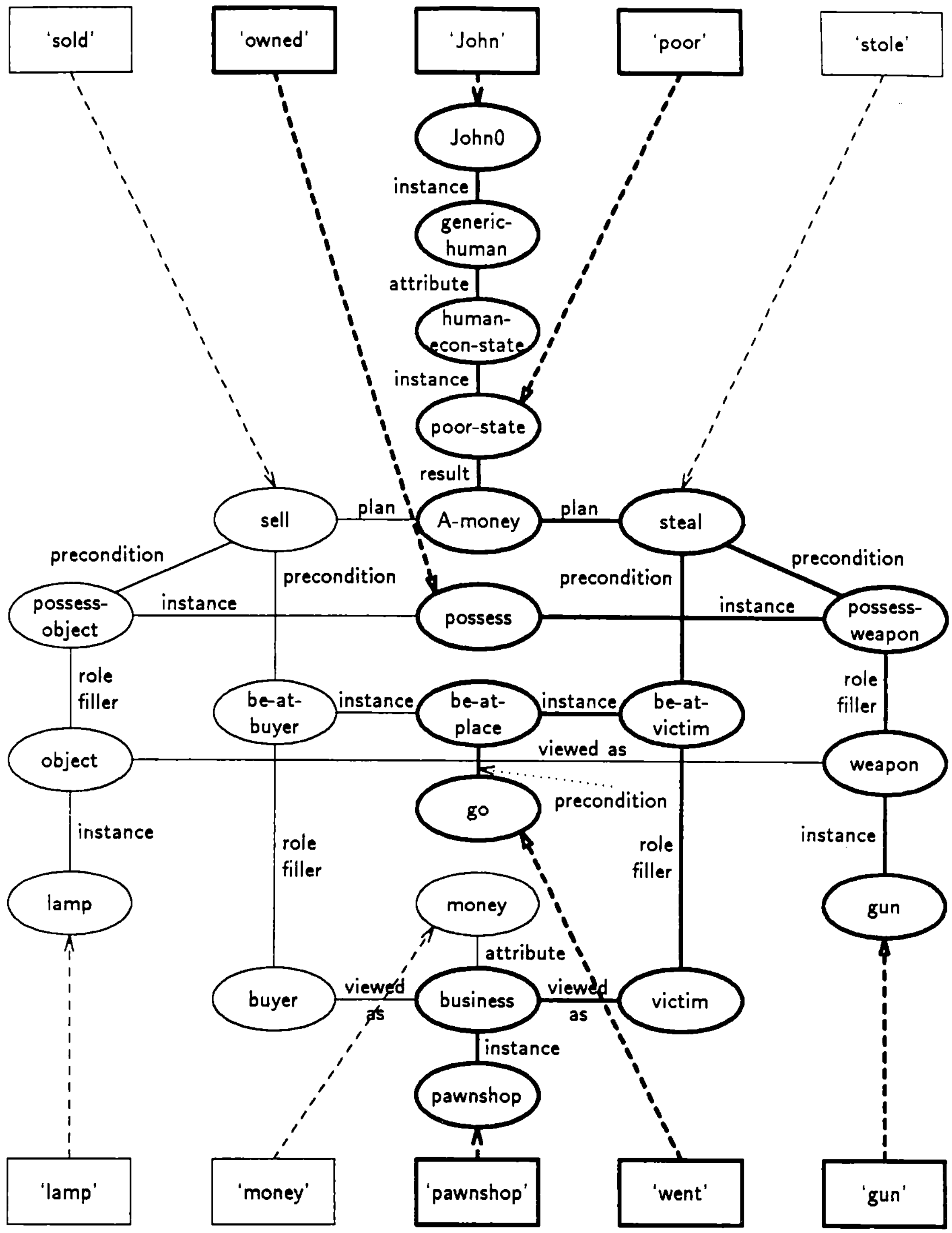

Figure A.2: Active paths in memory after processing the first three sentences of Text 10.

```
    SELL has the precondition POSSESS-OBJECT
    POSSESS-OBJECT is an instance of POSSESS
Also reconsidering (path2 path6 path4) due to shared nodes
  with path12
    Activating path12
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2
  More-reinforcement metric -- path8 has more shared nodes
    than path6
    De-activating path6
  More-reinforcement metric -- path3 has more shared nodes
    than path4
    De-activating path4

New path discovered: path13
  Path from SELL to GO
    SELL has the precondition BE-AT-BUYER
    BE-AT-BUYER is an instance of BE-AT-PLACE
    BE-AT-PLACE has the precondition GO
Also reconsidering (path6 path5 path9) due to shared nodes
  with path13
    Activating path13
  More-reinforcement metric -- path8 has more shared nodes
    than path6
    De-activating path6
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5
  More-reinforcement metric -- path10 has more shared nodes
    than path9
    De-activating path9
```

Now ATLAST adds path15 to its representation of the text, connecting John to the selling plan. Once this occurs there are ten active paths in ATLAST's interpretation of the story: five paths supporting the stealing interpretation, four supporting the selling interpretation, and one neutral path. While there are more paths favoring stealing, the four paths representing the selling plan are relatively short paths. Any yet-to-be-discovered paths which compete with these four paths and also incorporate the stealing plan will be longer than their competitors (and fairly uninformative because they include is-a intersections), as in the case of path17 below which competes with and loses to path12. The longer paths will lose when evaluated against the shorter paths and the four paths supporting the selling plan will remain active.

On the other hand, the five active paths which support the stealing plan compete with five retained paths which support the selling plan. These retained paths are approximately the same length as their competitors (give or take a viewed-as link, which is disregarded in assessing path length) and will consistently win when re-evaluated because of the reinforcement provided by the four short and effectively immovable paths supporting the selling plan. An example of this occurs below during the processing of path18 in which the retained path5 is re-evaluated and supplants path7. Thus, as new paths are evaluated and retained paths are thereby re-evaluated, ATLAST will incrementally supplant its "steal money" interpretation with the "sell possessions" interpretation.

```
New path discovered: path15
  Path from SELL to JOHNO
    SELL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
    POOR-STATE is an instance of HUMAN-ECON-STATE
    HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance JOHNO
Also reconsidering (path2 path5 path14) due to shared nodes
  with path15
    Activating path15
  More-reinforcement metric -- path1 has more shared nodes
    than path2
    De-activating path2
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5
  Shorter-path metric -- path12 shorter than path14
    De-activating path14
```

```
New path discovered: path17
  Path from SELL to POSSESS
    SELL has the precondition POSSESS-OBJECT
    POSSESS-OBJECT has the role-filler OBJECT
    OBJECT can be viewed as WEAPON
    WEAPON is a role-filler of POSSESS-WEAPON
    POSSESS-WEAPON is an instance of POSSESS
Also reconsidering (path6 path14 path2 path4) due to shared nodes
  with path17
  Shorter-path metric -- path12 shorter than path17
    De-activating path17
          :
          :

New path discovered: path18
  Path from SELL to POOR-STATE
    SELL has the precondition POSSESS-OBJECT
    POSSESS-OBJECT is an instance of POSSESS
    POSSESS has the instance POSSESS-WEAPON
    POSSESS-WEAPON is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
Also reconsidering (path6 path14 path2 path5 path16 path17 path4)
  due to shared nodes with path18
          :
          :
  No-decision metric -- path1 and path2 are retained
    De-activating path1
    De-activating path2
  More-reinforcement metric -- path5 has more shared nodes
    than path7
    De-activating path7
    Activating path5
          :
          :

Old path rediscovered: path17
Also reconsidering (path4 path2 path1 path14 path6 path18) due to
  shared nodes with path17
  Shorter-path metric -- path12 shorter than path17
    De-activating path17
  More-specific metric -- path3 more specific than path4
    De-activating path4
```

```
    Activating path2
  More-reinforcement metric -- path2 has more shared nodes
    than path1
    De-activating path1
  Shorter-path metric -- path12 shorter than path14
    De-activating path14
  More-reinforcement metric -- path6 has more shared nodes
    than path8
    De-activating path8
    Activating path6
  Shorter-path metric -- path11 shorter than path18
    De-activating path18

New path discovered: path19
  Path from SELL to POOR-STATE
    SELL has the precondition BE-AT-BUYER
    BE-AT-BUYER is an instance of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-VICTIM
    BE-AT-VICTIM is a precondition of STEAL
    STEAL is a plan of A-MONEY
    A-MONEY is a result of POOR-STATE
Also reconsidering (path18 path8 path14 path1 path16 path7 path9)
  due to shared nodes with path19
          :
          :
  More-reinforcement metric -- path9 has more shared nodes
    than path10
    De-activating path10
    Activating path9

Active memory structure:

  Paths: (path9 path6 path2 path5 path15 path13 path12 path11
          path3 path0)
  Path from PAWNSHOP to GO
    PAWNSHOP is an instance of BUSINESS
    BUSINESS can be viewed as BUYER
    BUYER is a role-filler of BE-AT-BUYER
    BE-AT-BUYER is an instance of BE-AT-PLACE
    BE-AT-PLACE has the precondition GO
  Path from GO to POSSESS
    GO is a precondition of BE-AT-PLACE
    BE-AT-PLACE has the instance BE-AT-BUYER
```

```
  BE-AT-BUYER is a precondition of SELL
  SELL has the precondition POSSESS-OBJECT
  POSSESS-OBJECT is an instance of POSSESS
Path from POSSESS to POOR-STATE
  POSSESS has the instance POSSESS-OBJECT
  POSSESS-OBJECT is a precondition of SELL
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
Path from GO to POOR-STATE
  GO is a precondition of BE-AT-PLACE
  BE-AT-PLACE has the instance BE-AT-BUYER
  BE-AT-BUYER is a precondition of SELL
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
Path from SELL to JOHNO
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
  POOR-STATE is an instance of HUMAN-ECON-STATE
  HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
  GENERIC-HUMAN has the instance JOHNO
Path from SELL to GO
  SELL has the precondition BE-AT-BUYER
  BE-AT-BUYER is an instance of BE-AT-PLACE
  BE-AT-PLACE has the precondition GO
Path from SELL to POSSESS
  SELL has the precondition POSSESS-OBJECT
  POSSESS-OBJECT is an instance of POSSESS
Path from SELL to POOR-STATE
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
Path from GUN to POSSESS
  GUN is an instance of WEAPON
  WEAPON is a role-filler of POSSESS-WEAPON
  POSSESS-WEAPON is an instance of POSSESS
Path from POOR-STATE to JOHNO
  POOR-STATE is an instance of HUMAN-ECON-STATE
  HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
  GENERIC-HUMAN has the instance JOHNO
```

ATLAST's interpretation now consists of ten paths. With one exception, all paths either support the "sell possession" interpretation or are neutral (i.e., path0). The exception is path3, which connects GUN to POSSESS and is the only active path containing the GUN node. As the remainder of the final sentence is read, ATLAST is reminded of the retained path4 which also includes the GUN node and makes the final correction. The system also adds a path connecting GUN and SELL.

```
Old path rediscovered: path4
Also reconsidering (path17 path18) due to shared nodes with path4
  More-reinforcement metric -- path4 has more shared nodes
    than path3
    De-activating path3
    Activating path4
          :
          :

New path discovered: path21
  Path from GUN to SELL
    GUN is an instance of WEAPON
    WEAPON can be viewed as OBJECT
    OBJECT is a role-filler of POSSESS-OBJECT
    POSSESS-OBJECT is a precondition of SELL
Also reconsidering (path18 path17 path3) due to shared nodes
  with path21
    Activating path21
          :
          :
```

ATLAST completes processing of Text 10 and issues its final interpretation:

```
Processing completed

  Active memory structure:

    Paths: (path21 path4 path9 path6 path2 path5 path15 path13
           path12 path11 path0)
    Path from GUN to SELL
      GUN is an instance of WEAPON
      WEAPON can be viewed as OBJECT
      OBJECT is a role-filler of POSSESS-OBJECT
      POSSESS-OBJECT is a precondition of SELL
    Path from GUN to POSSESS
      GUN is an instance of WEAPON
      WEAPON can be viewed as OBJECT
      OBJECT is a role-filler of POSSESS-OBJECT
```

```
  POSSESS-OBJECT is an instance of POSSESS
Path from PAWNSHOP to GO
  PAWNSHOP is an instance of BUSINESS
  BUSINESS can be viewed as BUYER
  BUYER is a role-filler of BE-AT-BUYER
  BE-AT-BUYER is an instance of BE-AT-PLACE
  BE-AT-PLACE has the precondition GO
Path from GO to POSSESS
  GO is a precondition of BE-AT-PLACE
  BE-AT-PLACE has the instance BE-AT-BUYER
  BE-AT-BUYER is a precondition of SELL
  SELL has the precondition POSSESS-OBJECT
  POSSESS-OBJECT is an instance of POSSESS
Path from POSSESS to POOR-STATE
  POSSESS has the instance POSSESS-OBJECT
  POSSESS-OBJECT is a precondition of SELL
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
Path from GO to POOR-STATE
  GO is a precondition of BE-AT-PLACE
  BE-AT-PLACE has the instance BE-AT-BUYER
  BE-AT-BUYER is a precondition of SELL
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
Path from SELL to JOHNO
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
  POOR-STATE is an instance of HUMAN-ECON-STATE
  HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
  GENERIC-HUMAN has the instance JOHNO
Path from SELL to GO
  SELL has the precondition BE-AT-BUYER
  BE-AT-BUYER is an instance of BE-AT-PLACE
  BE-AT-PLACE has the precondition GO
Path from SELL to POSSESS
  SELL has the precondition POSSESS-OBJECT
  POSSESS-OBJECT is an instance of POSSESS
Path from SELL to POOR-STATE
  SELL is a plan of A-MONEY
  A-MONEY is a result of POOR-STATE
Path from POOR-STATE to JOHNO
  POOR-STATE is an instance of HUMAN-ECON-STATE
```

```
    HUMAN-ECON-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance JOHNO

Pointers to memory structure:

  Event:  event0
  Actor:  (JOHNO)
  Action: (BE)
  Object: (POOR-STATE)
  Direction: nil

  Event:  event1
  Actor:  (JOHNO)
  Action: (POSSESS)
  Object: (GUN)
  Direction: nil

  Event:  event2
  Actor:  (JOHNO)
  Action: (GO)
  Object: nil
  Direction: (PAWNSHOP)

  Event:  event3
  Actor:  (JOHNO)
  Action: (SELL)
  Object: (GUN)
  Direction: nil
```

ATLAST's final interpretation of Text 10 is, in part, a network of relationships between concepts represented by the 11 paths listed above. This network is highlighted in Figure A.3. In the process of deciding upon those 11 paths, ATLAST discovered and ultimately rejected 13 others. This example has concentrated on the path evaluation process. However, the collection of chosen paths alone does not provide a complete understanding of the text. Thus, concurrent with the discovery and selection of explanatory inference paths, ATLAST has constructed a sequence of pointers into the network which provides a temporal ordering on the events and a determination of thematic role assignments (i.e., who did what and when).

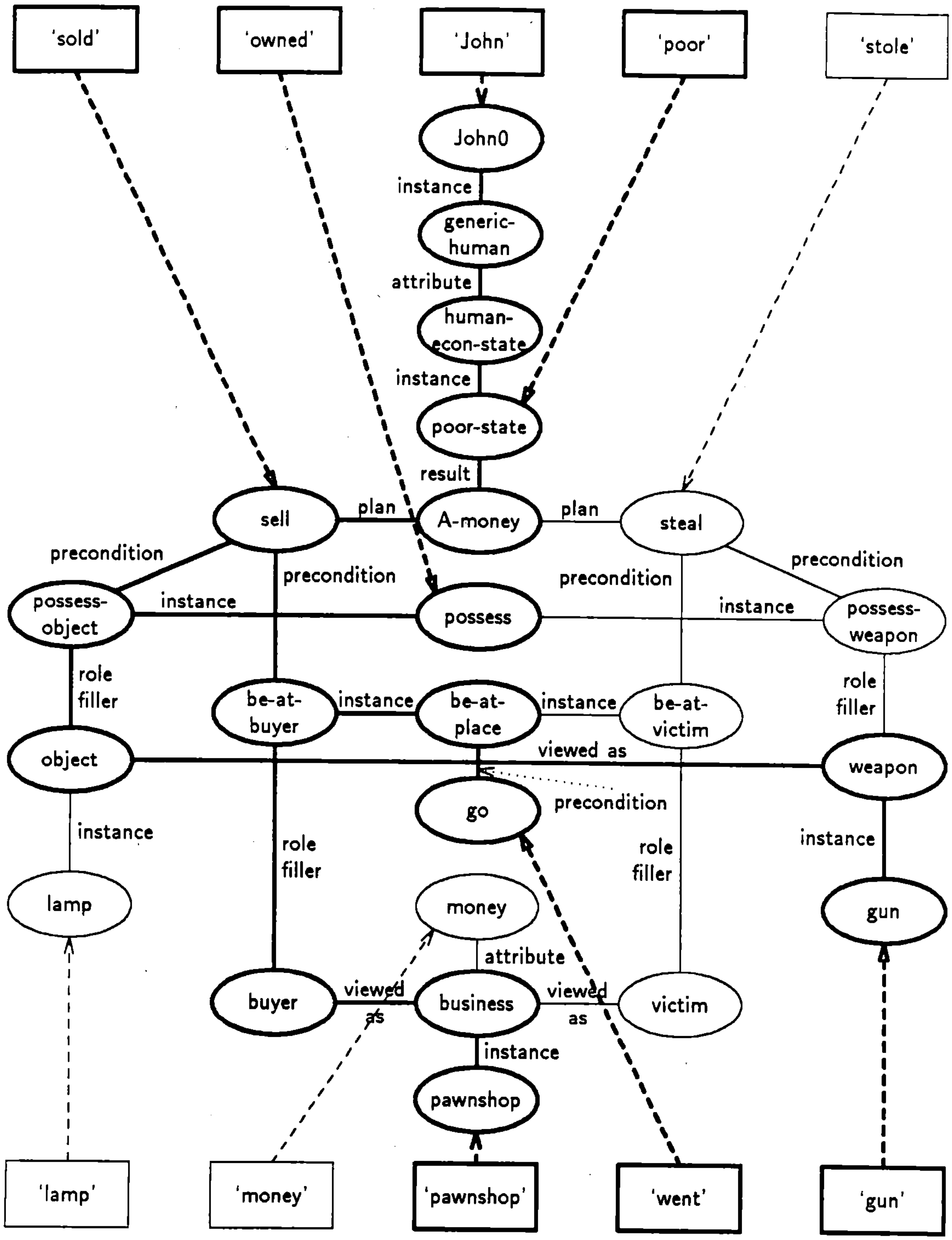

Figure A.3: Active paths in memory after processing all four sentences of Text 10.

In brief, ATLAST has resolved the pragmatic ambiguity presented by Text 10, incorrectly at first, and then corrected its mistake. This is accomplished using the same processes used in the lexical ambiguity example of Chapter 5. There are no changes to the program other than the two parameter values mentioned at the beginning of this example.

A.2 Strategy-driven inference processing

In the example above, ATLAST's ability to process the text successfully stems from two fundamental assumptions: (1) the inseparability and uniform representation of lexical and pragmatic knowledge, and (2) the retention of rejected inferences. The examples which follow will demonstrate how those same two assumptions enable ATLAST to model the strategy-driven inference behavior described in Chapter 6. The first of these examples will show ATLAST modeling perseverer behavior on a simplified version of Text 8:

Text 8: Wilma began to cry.
Fred had just asked her to marry him.

The second example will show how ATLAST models recency behavior and arrives at a different interpretation of the same text due only to a difference in how ATLAST deals with unresolvable inference decisions. The semantic memory used in these two examples is shown in Figure A.4.

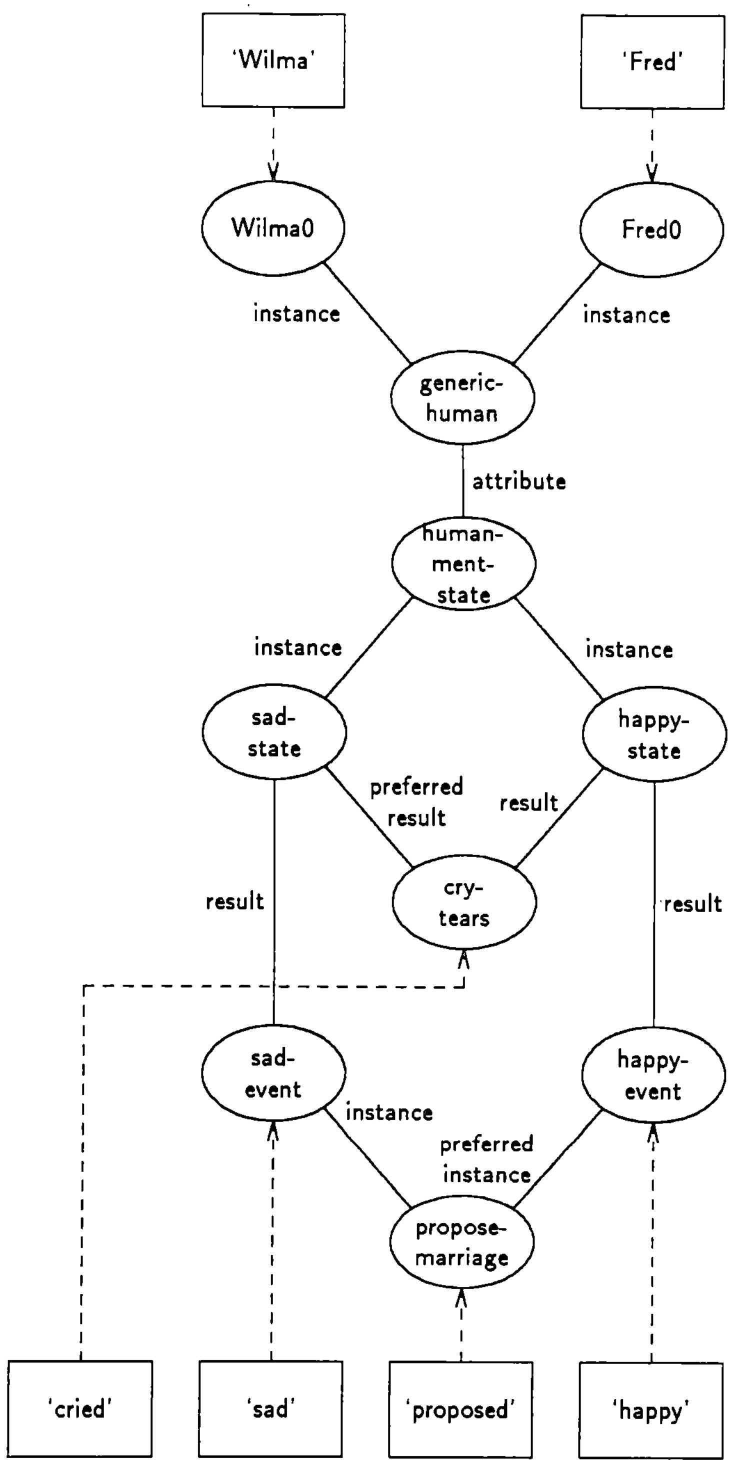

Figure A.4: Memory network for Text 11.

A.2.1 An example of perseverer behavior

As with the previous examples, the input text has been simplified for the next two examples. The simplified text is:

Text 11: Wilma cried.
Fred had proposed to Wilma.

As ATLAST begins processing this example using the perseverer strategy, a new inference evaluation metric, the preferred-link metric, is added to those used in previous examples. The parameters controlling retention and re-evaluation again have been tuned for better performance.

```
Input text is:

    Wilma cried.
    Fred had proposed to Wilma.

Inference strategy in effect: perseverer

Ordering of inference evaluation metrics in force:

    MORE-ACTIVATION-METRIC
    SHORTER-PATH-METRIC
    MORE-REINFORCEMENT-METRIC
    MORE-SPECIFIC-METRIC
    PREFERRED-LINK-METRIC
    NO-DECISION-METRIC

Maximum distance of marker-passing: 3

Distance to pass markers per cycle: 3
Are rejected paths being retained?: t
Max. no. of unsuccessful evaluations: 3
Min. no. of shared nodes to force re-evaluation: 4

Are function words being allocated processing cycles?: nil
Is inference processing forced to complete at periods?: t
```

During the processing of the first sentence, ATLAST finds competing paths joining the words `Wilma` and `cried`.

```
New path discovered: path0
  Path from CRY-TEARS to WILMA0
    CRY-TEARS is a result of HAPPY-STATE
    HAPPY-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance WILMA0
    Activating path0

New path discovered: path1
  Path from CRY-TEARS to WILMA0
    CRY-TEARS is a preferred result of SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance WILMA0
  Preferred-link metric -- path1 has more pref links than path0
    De-activating path0
    Activating path1
```

The first four inference evaluation metrics are unable to make a decision between the competing paths. This would normally result in a split-decision to be resolved later, but in this case one of the paths, path1, contains a **preferred** link. The preferred-link metric determines therefore that path1 is a default inference and chooses it over path0. Because the preferred-link metric is always the last decision-making metric to be invoked, ATLAST uses default inferences only as a last resort.

```
Active memory structure:

  Paths: (path1)
  Path from CRY-TEARS to WILMA0
    CRY-TEARS is a preferred result of SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance WILMA0
```

```
Pointers to memory structure:

  Event:   event0
  Actor:   (WILMA0)
  Action: (CRY-TEARS)
  Object: nil
  Direction: nil
```

As ATLAST processes the second sentence, any inference decisions are biased by the active context from the first sentence—that Wilma is crying because she is sad. Thus, Fred's marriage proposal, which would normally be viewed as a happy event, is interpreted as one causing despair.

```
New path discovered: path2
  Path from PROPOSE-MARRIAGE to CRY-TEARS
    PROPOSE-MARRIAGE is a preferred instance of HAPPY-EVENT
    HAPPY-EVENT has the result HAPPY-STATE
    HAPPY-STATE has the result CRY-TEARS
    Activating path2

New path discovered: path3
  Path from PROPOSE-MARRIAGE to CRY-TEARS
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE has the preferred result CRY-TEARS
  More-reinforcement metric -- path3 has more shared nodes
    than path2
    De-activating path2
    Activating path3

New path discovered: path5
  Path from PROPOSE-MARRIAGE to FRED0
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FRED0
Also reconsidering (path4) due to shared nodes with path5
    Activating path5
  Shorter-path metric -- path3 shorter than path4
    De-activating path4
```

```
New path discovered: path7
  Path from PROPOSE-MARRIAGE to FREDO
    PROPOSE-MARRIAGE is a preferred instance of HAPPY-EVENT
    HAPPY-EVENT has the result HAPPY-STATE
    HAPPY-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FREDO
  More-reinforcement metric -- path5 has more shared nodes
    than path7
    De-activating path7

Active memory structure:

  Paths: (path5 path3 path1)
  Path from PROPOSE-MARRIAGE to FREDO
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FREDO
  Path from PROPOSE-MARRIAGE to CRY-TEARS
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE has the preferred result CRY-TEARS
  Path from CRY-TEARS to WILMAO
    CRY-TEARS is a preferred result of SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance WILMAO
```

ATLAST reads the last two words of the text, `to Wilma`, and finds two more paths that reinforce its existing interpretation.

```
New path discovered: path8
  Path from WILMAO to FREDO
    WILMAO is an instance of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FREDO
    Activating path8

New path discovered: path9
  Path from WILMAO to PROPOSE-MARRIAGE
    WILMAO is an instance of GENERIC-HUMAN
    GENERIC-HUMAN has the attribute HUMAN-MENT-STATE
```

```
    HUMAN-MENT-STATE has the instance SAD-STATE
    SAD-STATE is a result of SAD-EVENT
    SAD-EVENT has the instance PROPOSE-MARRIAGE
Also reconsidering (path4) due to shared nodes with path9
    Activating path9
  Shorter-path metric -- path3 shorter than path4
    De-activating path4
    Suppressing path4

New path discovered: path10
  Path from WILMA0 to PROPOSE-MARRIAGE
    WILMA0 is an instance of GENERIC-HUMAN
    GENERIC-HUMAN has the attribute HUMAN-MENT-STATE
    HUMAN-MENT-STATE has the instance HAPPY-STATE
    HAPPY-STATE is a result of HAPPY-EVENT
    HAPPY-EVENT has the preferred instance PROPOSE-MARRIAGE
Also reconsidering (path7) due to shared nodes with path10
  More-reinforcement metric -- path9 has more shared nodes
    than path10
    De-activating path10
  More-reinforcement metric -- path5 has more shared nodes
    than path7
    De-activating path7
    Suppressing path7

Active memory structure:

  Paths: (path9 path8 path5 path3 path1)
  Path from WILMA0 to PROPOSE-MARRIAGE
    WILMA0 is an instance of GENERIC-HUMAN
    GENERIC-HUMAN has the attribute HUMAN-MENT-STATE
    HUMAN-MENT-STATE has the instance SAD-STATE
    SAD-STATE is a result of SAD-EVENT
    SAD-EVENT has the instance PROPOSE-MARRIAGE
  Path from WILMA0 to FRED0
    WILMA0 is an instance of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FRED0
  Path from PROPOSE-MARRIAGE to FRED0
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FRED0
```

```
    Path from PROPOSE-MARRIAGE to CRY-TEARS
      PROPOSE-MARRIAGE is an instance of SAD-EVENT
      SAD-EVENT has the result SAD-STATE
      SAD-STATE has the preferred result CRY-TEARS
    Path from CRY-TEARS to WILMAO
      CRY-TEARS is a preferred result of SAD-STATE
      SAD-STATE is an instance of HUMAN-MENT-STATE
      HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
      GENERIC-HUMAN has the instance WILMAO

Processing completed

  Pointers to memory structure:

    Event:  event1
    Actor:  (FREDO)
    Action: (PROPOSE-MARRIAGE)
    Object: nil
    Direction: (WILMAO)

    Event:  event0
    Actor:  (WILMAO)
    Action: (CRY-TEARS)
    Object: nil
    Direction: nil
```

This example of ATLAST's perseverer strategy provides another demonstration of ATLAST's ability to resolve pragmatic ambiguities using the same processes that were previously employed in a lexical disambiguation task. ATLAST's interpretation of Text 11 is displayed in Figure A.5. If the reader senses something familiar about the perseverer strategy, it is because the perseverer strategy was used in all previous examples. The preferred-link metric was not used during the processing of Text 5 or Text 10 so it was omitted from the sample output. In this sense, the example just presented is redundant in that it tells us nothing new about ATLAST. It is included here as a basis for comparison to ATLAST's processing of the same text, Text 11, using the recency strategy.

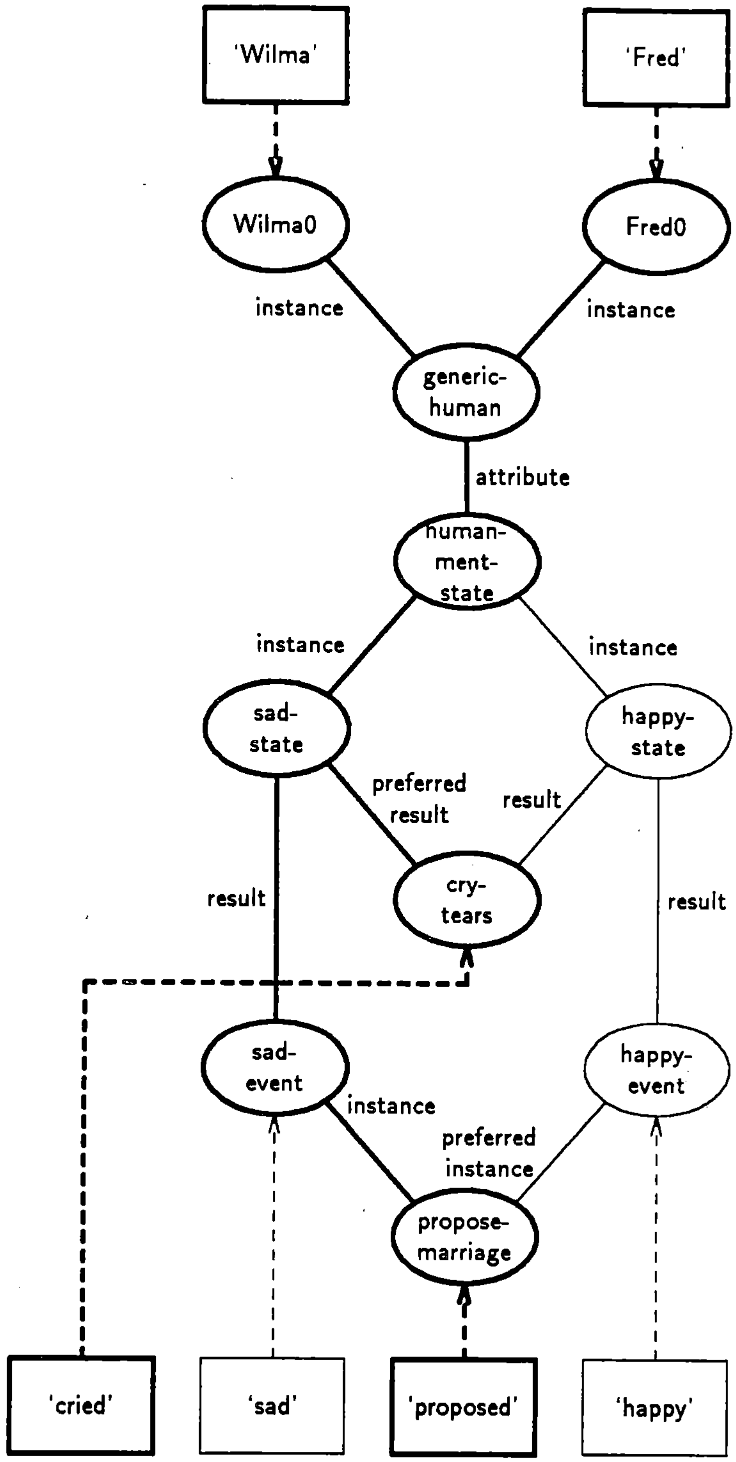

Figure A.5: ATLAST's *perseverer* interpretation of Text 11.

A.2.2 An example of recency behavior

ATLAST's recency strategy for inference processing is different in only two ways from its perseverer strategy. One difference is that the occurrence of an unresolved path competition results in both competing paths being locked out of the inference evaluation process until ATLAST has read the entire text. These postponed decisions are stored in a queue so that the most recently postponed decisions will be the first to be re-evaluated at the end of the text. Thus, the recency strategy postpones the resolution of split decisions, while the perseverer strategy resolves them as soon as possible. In conjunction with the forced delay in deciding ties, the recency strategy also does not invoke the tie-breaking rule, the preferred-link metric, until the end of the text when the postponed split decisions are reprocessed. This is the second difference between the the two strategies.

When told to use the recency strategy, ATLAST's inference processing changes only in the two ways just stated, yet because of these two differences, ATLAST will arrive at a different interpretation of Text 11 using the recency strategy than it did when using the perseverer strategy. All other parameters are the same as they were in the earlier perseverer example.

```
Input text is:

    Wilma cried.
    Fred had proposed to Wilma.

Inference strategy in effect: recency
```

Again ATLAST reads the sentence, `Wilma cried`. This time, however, the tie between path0 and path1 is not immediately resolved. The two paths are retained and prevented from being re-evaluated until the end of the text.

```
New path discovered: path0
  Path from CRY-TEARS to WILMA0
    CRY-TEARS is a result of HAPPY-STATE
    HAPPY-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance WILMA0
    Activating path0

New path discovered: path1
  Path from CRY-TEARS to WILMA0
    CRY-TEARS is a preferred result of SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance WILMA0
  No-decision metric -- path0 and path1 are retained
    De-activating path0
    De-activating path1

Old path rediscovered: path0
Also reconsidering (path1) due to tie with path0
  Evaluation of path0 postponed due to recency strategy
  Evaluation of path1 postponed due to recency strategy

Active memory structure:

  Paths: nil

Pointers to memory structure:

  Event:  event0
  Actor:  (WILMA0)
  Action: (CRY-TEARS)
  Object: nil
  Direction: nil
```

Because no decision was made during the processing of the first sentence, there is no active context to guide the understanding of the second sentence. As ATLAST discovers competing pairs of paths in this network, it is unable to make any decisions at all. ATLAST activates only path8, which simply says that Fred and Wilma are both human, by the end of the second sentence.

```
New path discovered: path2
  Path from PROPOSE-MARRIAGE to CRY-TEARS
    PROPOSE-MARRIAGE is a preferred instance of HAPPY-EVENT
    HAPPY-EVENT has the result HAPPY-STATE
    HAPPY-STATE has the result CRY-TEARS
    Activating path2

New path discovered: path3
  Path from PROPOSE-MARRIAGE to CRY-TEARS
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE has the preferred result CRY-TEARS
  No-decision metric -- path2 and path3 are retained
    De-activating path2
    De-activating path3

Old path rediscovered: path3
Also reconsidering (path2) due to tie with path3
  Evaluation of path3 postponed due to recency strategy
  Evaluation of path2 postponed due to recency strategy

New path discovered: path4
  Path from PROPOSE-MARRIAGE to CRY-TEARS
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE has the instance HAPPY-STATE
    HAPPY-STATE has the result CRY-TEARS
    Activating path4

New path discovered: path5
  Path from PROPOSE-MARRIAGE to FREDO
    PROPOSE-MARRIAGE is an instance of SAD-EVENT
    SAD-EVENT has the result SAD-STATE
    SAD-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FREDO
    Activating path5

New path discovered: path6
  Path from PROPOSE-MARRIAGE to CRY-TEARS
    PROPOSE-MARRIAGE is a preferred instance of HAPPY-EVENT
    HAPPY-EVENT has the result HAPPY-STATE
```

```
    HAPPY-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE has the instance SAD-STATE
    SAD-STATE has the preferred result CRY-TEARS
  No-decision metric -- path4 and path6 are retained
    De-activating path4
    De-activating path6

New path discovered: path7
  Path from PROPOSE-MARRIAGE to FREDO
    PROPOSE-MARRIAGE is a preferred instance of HAPPY-EVENT
    HAPPY-EVENT has the result HAPPY-STATE
    HAPPY-STATE is an instance of HUMAN-MENT-STATE
    HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FREDO
Also reconsidering (path6) due to shared nodes with path7
Also reconsidering (path4) due to tie with path6
  No-decision metric -- path5 and path7 are retained
    De-activating path5
    De-activating path7
  Evaluation of path6 postponed due to recency strategy
  Evaluation of path4 postponed due to recency strategy

New path discovered: path8
  Path from WILMAO to FREDO
    WILMAO is an instance of GENERIC-HUMAN
    GENERIC-HUMAN has the instance FREDO
    Activating path8

New path discovered: path9
  Path from WILMAO to PROPOSE-MARRIAGE
    WILMAO is an instance of GENERIC-HUMAN
    GENERIC-HUMAN has the attribute HUMAN-MENT-STATE
    HUMAN-MENT-STATE has the instance SAD-STATE
    SAD-STATE is a result of SAD-EVENT
    SAD-EVENT has the instance PROPOSE-MARRIAGE
Also reconsidering (path1 path4 path5) due to shared nodes
  with path9
Also reconsidering (path0) due to tie with path1
Also reconsidering (path6) due to tie with path4
Also reconsidering (path7) due to tie with path5
    Activating path9
  Evaluation of path1 postponed due to recency strategy
  Evaluation of path4 postponed due to recency strategy
```

```
    Evaluation of path5 postponed due to recency strategy
    Evaluation of path0 postponed due to recency strategy
    Evaluation of path6 postponed due to recency strategy
    Evaluation of path7 postponed due to recency strategy

  New path discovered: path10
    Path from WILMA0 to PROPOSE-MARRIAGE
      WILMA0 is an instance of GENERIC-HUMAN
      GENERIC-HUMAN has the attribute HUMAN-MENT-STATE
      HUMAN-MENT-STATE has the instance HAPPY-STATE
      HAPPY-STATE is a result of HAPPY-EVENT
      HAPPY-EVENT has the preferred instance PROPOSE-MARRIAGE
  Also reconsidering (path0 path6 path7) due to shared nodes
    with path10
  Also reconsidering (path1) due to tie with path0
  Also reconsidering (path4) due to tie with path6
  Also reconsidering (path5) due to tie with path7
    No-decision metric -- path9 and path10 are retained
      De-activating path9
      De-activating path10
    Evaluation of path0 postponed due to recency strategy
    Evaluation of path6 postponed due to recency strategy
    Evaluation of path7 postponed due to recency strategy
    Evaluation of path1 postponed due to recency strategy
    Evaluation of path4 postponed due to recency strategy
    Evaluation of path5 postponed due to recency strategy

  Active memory structure:

    Paths: (path8)
    Path from WILMA0 to FRED0
      WILMA0 is an instance of GENERIC-HUMAN
      GENERIC-HUMAN has the instance FRED0
```

Having reached the end of the text, ATLAST invokes the preferred-link metric to aid in resolving the postponed decisions. The system then begins to re-evaluate the postponed decisions beginning with the most recently postponed.

```
Revised ordering of inference evaluation metrics in force:

    MORE-ACTIVATION-METRIC
    SHORTER-PATH-METRIC
```

```
    MORE-REINFORCEMENT-METRIC
    MORE-SPECIFIC-METRIC
    PREFERRED-LINK-METRIC
    NO-DECISION-METRIC

Now processing postponed evaluations

  Postponed path being evaluated: path10
    Also reconsidering: (path6 path7 path0)
      Activating path10
      Activating path6
      Activating path7
      Activating path0

  Postponed path being evaluated: path9
    Also reconsidering: (path4 path5 path1)
    Preferred-link metric -- path10 has more pref links than path9
      De-activating path9
    More-reinforcement metric -- path6 has more shared nodes
      than path4
      De-activating path4
    More-reinforcement metric -- path7 has more shared nodes
      than path5
      De-activating path5
    More-reinforcement metric -- path0 has more shared nodes
      than path1
      De-activating path1

  Postponed path being evaluated: path5
    Also reconsidering: (path4 path9)
    More-reinforcement metric -- path7 has more shared nodes
      than path5
      De-activating path5
    More-reinforcement metric -- path6 has more shared nodes
      than path4
      De-activating path4
    More-reinforcement metric -- path10 has more shared nodes
      than path9
      De-activating path9

  Postponed path being evaluated: path4
    Also reconsidering: (path9 path5)
```

```
  More-reinforcement metric -- path6 has more shared nodes
    than path4
    De-activating path4
  More-reinforcement metric -- path10 has more shared nodes
    than path9
    De-activating path9
  More-reinforcement metric -- path7 has more shared nodes
    than path5
    De-activating path5

Postponed path being evaluated: path3
  Shorter-path metric -- path3 shorter than path6
    De-activating path6
    Activating path3

Postponed path being evaluated: path2
  More-reinforcement metric -- path2 has more shared nodes
    than path3
    De-activating path3
    Activating path2

Postponed path being evaluated: path1
  Also reconsidering: (path9)
  More-reinforcement metric -- path0 has more shared nodes
    than path1
    De-activating path1
  More-reinforcement metric -- path10 has more shared nodes
    than path9
    De-activating path9
    Suppressing path9
```

In this example, every one of ATLAST's inference decisions has been initially postponed and later re-evaluated. Because the first postponed decisions to be re-evaluated were spawned from Fred's marriage proposal and the proposal is regarded out of context as a happy event, ATLAST has explained Wilma's tears as tears of joy. This explanation of Text 11, which is shown in Figure A.6, differs greatly from that obtained by the perseverer strategy.

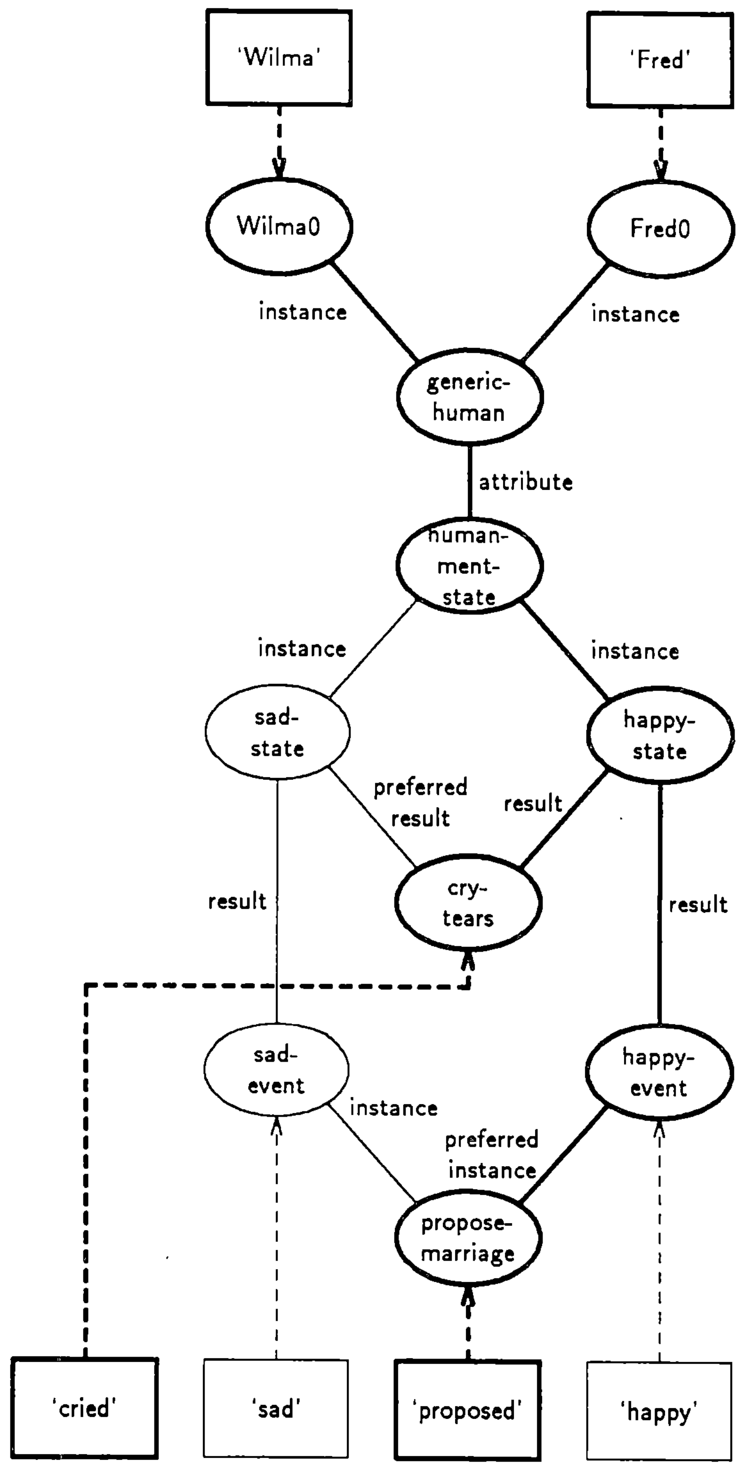

Figure A.6: ATLAST's *recency* interpretation of Text 11.

```
Processing completed

 Active memory structure:

   Paths: (path2 path0 path7 path10 path8)
   Path from PROPOSE-MARRIAGE to CRY-TEARS
     PROPOSE-MARRIAGE is a preferred instance of HAPPY-EVENT
     HAPPY-EVENT has the result HAPPY-STATE
     HAPPY-STATE has the result CRY-TEARS
   Path from CRY-TEARS to WILMA0
     CRY-TEARS is a result of HAPPY-STATE
     HAPPY-STATE is an instance of HUMAN-MENT-STATE
     HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
     GENERIC-HUMAN has the instance WILMA0
   Path from PROPOSE-MARRIAGE to FRED0
     PROPOSE-MARRIAGE is a preferred instance of HAPPY-EVENT
     HAPPY-EVENT has the result HAPPY-STATE
     HAPPY-STATE is an instance of HUMAN-MENT-STATE
     HUMAN-MENT-STATE is an attribute of GENERIC-HUMAN
     GENERIC-HUMAN has the instance FRED0
   Path from WILMA0 to PROPOSE-MARRIAGE
     WILMA0 is an instance of GENERIC-HUMAN
     GENERIC-HUMAN has the attribute HUMAN-MENT-STATE
     HUMAN-MENT-STATE has the instance HAPPY-STATE
     HAPPY-STATE is a result of HAPPY-EVENT
     HAPPY-EVENT has the preferred instance PROPOSE-MARRIAGE
   Path from WILMA0 to FRED0
     WILMA0 is an instance of GENERIC-HUMAN
     GENERIC-HUMAN has the instance FRED0

 Pointers to memory structure:

   Event:  event1
   Actor:  (FRED0)
   Action: (PROPOSE-MARRIAGE)
   Object: nil
   Direction: (WILMA0)

   Event:  event0
   Actor:  (WILMA0)
   Action: (CRY-TEARS)
   Object: nil
   Direction: nil
```

A.2.3 Comparing ATLAST's behavior to human behavior

Granger and Holbrook's (1983) original work on strategy-driven inference behavior included a test of subjects' preferences to determine the default inferences associated with individual story events. As reported in Chapter 6, Granger and Holbrook found that, in the absence of biasing context, subjects consistently inferred that crying was associated with being sad or upset while a marriage proposal was associated with joy or happiness. ATLAST exhibits this same behavior: when presented with either of the sentences of Text 11 in isolation, ATLAST will make the default or preferred decision regardless of the inference strategy in effect. The only difference is that, when using the perseverer strategy, the system will make the default decision almost immediately, while it will wait until the end of the sentence when using the recency strategy.

Granger and Holbrook also noted that differences in behavior were made obvious only through the use of specially constructed, reciprocally ambiguous stories. In most other cases, they said, readers using different strategies will arrive at the same interpretation of story events. This effect is modeled by ATLAST as well: ATLAST will derive the same interpretation for Text 5 using either the perseverer or recency strategy, and the same holds true for Text 10.

Finally, it was stated in Chapter 6 that retention of competing inferences was at the heart of strategy-driven inference processing. More specifically, the recency strategy depends on inference retention. If rejected paths are subsequently discarded without any trace, the inference processor using the recency strategy will have no way of tracking the postponed decisions. The competing paths involved in those decisions will be indistinguishable from the rest of inactive memory. This is what happens when ATLAST is using the recency strategy on Text 11 but inference retention is disabled.

ATLAST processes the text, postponing split decisions along the way, but when it reaches the end of the text and attempts to reprocess those postponed decisions it no longer knows which paths were involved. The resulting interpretation is little more than that Fred and Wilma are both human. On the other hand, ATLAST is able to find the correct interpretation using the perseverer strategy with no inference retention because Text 11 does not mislead the system. If Text 11 did require error recovery, ATLAST would be in trouble. Thus, while both strategies need conditional retention for error recovery, the recency strategy relies on conditional retention even for normal processing without errors.

Appendix B

The Proof

We wish to show that for a network of n nodes in which every pair of nodes is joined by a single bi-directional link, the number of paths (P) that join two arbitrary nodes and have length less than or equal to l is given by the equation:

$$P = \sum_{i=1}^{l} \frac{(n-2)!}{(n-i-1)!}$$

or, in a simpler form, that the number of paths that join two arbitrary nodes and have length of exactly i is:

$$\frac{(n-2)!}{(n-i-1)!}$$

Definition: A *clique* of size n is an undirected graph G with vertices V and edges E such that the number of vertices in V is n and for all a and b in V such that $a \neq b$ there exists an edge (a, b) in E.

Definition: A *simple path* is a path which does not intersect itself.

Theorem: Given a clique of size n and two arbitrary vertices a and b in G with $a \neq b$, we wish to show that the number of distinct simple paths of length i, denoted $l_{i,n}$, is given by:

$$l_{i,n} = (n-2)(n-3)\cdots(n-i+1)(n-i) = \frac{(n-2)!}{(n-i-1)!}$$

Proof: By double induction.

Basis: Show that $l_{1,2} = 1$. This is trivial, since if a and b are two vertices in G, there is exactly one path of length 1 between them, which is the edge (a,b).

Inductive step 1: Assume the hypothesis is true for $i \leq I$ and $n \leq N$. Show that it is true for $i = I+1$ and $n = N$.

From vertex a in V, we form paths of length I to some intermediate vertex c, then add the edge from c to b, thereby forming a path from a to b of length $I+1$. There are $N-2$ intermediate vertices available, namely all vertices in V except a and b. We must now determine the number of simple paths of length I from a to each fixed c. These paths cannot go through b, since we would no longer have a simple path when we added the edge from c to b. This means that the the clique to be considered has size $N-1$. The number of simple paths of length $I+1$ from a to b then is the number of possible edges from b to some intermediate c multiplied by the number of paths of length I from a to c:

$$\begin{aligned}
l_{I+1,N} &= (N-2)\cdot l_{I,N-1} \\
&= (N-2)\cdot\frac{(N-3)!}{(N-1-I-1)!} \\
&= \frac{(N-2)!}{(N-I-2)!} \\
&= \frac{(n-2)!}{(n-(i-1)-2)!} \\
&= \frac{(n-2)!}{(n-i-1)!}
\end{aligned}$$

Inductive step 2: Assume the hypothesis is true for $i \leq I$ and $n \leq N$. Show that it is true for $i = I$ and $n = N + 1$.

From vertex a in V, we now form paths of length $I - 1$ to some intermediate vertex c and add the edge from c to b, giving paths from a to b of length I. There are $N - 1$ intermediate vertices available, since there are $N + 1$ vertices altogether. The number of simple paths of length $I - 1$ from a to a fixed intermediate vertex c is this time governed by a clique of size N as the paths again cannot go through b. There are $l_{I-1,N}$ of these paths and $N - 1$ edges from intermediate vertices to b, so the number of simple paths of length I from a to b is given by:

$$\begin{aligned} l_{I,N+1} &= (N-1) \cdot l_{I-1,N} \\ &= (N-1) \cdot \frac{(N-2)!}{(N-(I-1)-1)!} \\ &= \frac{(N-1)!}{(N-I)!} \\ &= \frac{(n-1-1)!}{(n-1-i)!} \\ &= \frac{(n-2)!}{(n-i-1)!} \end{aligned}$$

The proof is now complete. □