Knowledge-Based Contextual Reference Resolution for Text Understanding

Michael Kavanagh Smith

AI-TR-85-02 January 1985
# Table of Contents

1. Introduction
   1.1 Outline of remainder

2. Overview
   2.1 Context
   2.2 Aspects of reference
      2.2.1 Explicit vs. implicit reference
      2.2.2 Specificity of reference
      2.2.3 Lexical sense in context
      2.2.4 Preference ordering of candidate references
   2.3 The effects of reference resolution
   2.4 Summary

3. Representation Issues
   3.1 Levels of representation
      3.1.1 Physical level
      3.1.2 Program level
      3.1.3 Formal level
      3.1.4 Instances of representations
   3.2 Representation interpretation
   3.3 Representation overview
   3.4 Required properties
      3.4.1 Links
         3.4.1.1 ISA
         3.4.1.2 EQUIV, the manifestation link
         3.4.1.3 ROLE, the subpart mapping link
         3.4.1.4 The procedural links, *A* and NEQ
   3.4.2 Uniformity of reference across the different hierarchic relations.
   3.4.3 Merging of word classes in concepts
   3.4.4 Features on concepts
   3.5 System partitions
      3.5.1 World knowledge
      3.5.2 Workspace
      3.5.3 Context
      3.5.4 Lexicon
   3.6 Limitations of representation

4. Detailed Example
   4.1 Index references
      4.1.1 New instance
      4.1.2 Existing instance
      4.1.3 Implicit instance
   4.2 Reference by a more general term
      4.2.1 Existing instance
      4.2.2 Subpart of existing instance
   4.3 Reference by a specializing term
      4.3.1 Implicit instance
      4.3.2 Specialization of explicit concept by subpart
   4.4 Computed result of example sentence

5. Reference Procedures
   5.1 Input
5.2 Reference candidate search
  5.2.1 Tracing lexical senses
  5.2.2 Usage in context
    5.2.2.1 Superordinate usages
    5.2.2.2 Usage matching
  5.2.3 Embedded referents
  5.2.4 Reference candidates
5.3 Preference ordering of candidates
5.4 Reference testing
5.5 Representation modification
  5.5.1 Activation during instantiation
5.6 Noun phrase integration
6. Summary of Results
  6.1 The trading voyage schema
  6.2 The want and give schemas
  6.3 The flight schema
  6.4 The legal suit schema
7. Relation to Other Work
  7.1 Reference and context
    7.1.1 Lockman and Klappholz
    7.1.2 Grosz, Robinson, and Sidner
    7.1.3 Webber
    7.1.4 Clark
    7.1.5 The Yale group
  7.2 Lexical sense selection
  7.3 Coherence and text grammar
  7.4 Representation
8. Future Work and Conclusions
  8.1 The structure of context
  8.2 Intervening frame problem and schema selection
  8.3 Representation
  8.4 Parsing
  8.5 The representation of lexical senses
  8.6 Conclusion
I. Database for the two real texts
Bibliography
List of Figures

<table>
<thead>
<tr>
<th>Figure 3-1:</th>
<th>Levels of abstraction in cognitive theories</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3-2:</td>
<td>Representation requirements for contextual reference resolution</td>
<td>16</td>
</tr>
<tr>
<td>Figure 3-3:</td>
<td>Different relations implied by the ISA link depending on the features of connected concepts. (Note: CI = conceptual instance)</td>
<td>17</td>
</tr>
<tr>
<td>Figure 3-4:</td>
<td>CI definition example</td>
<td>18</td>
</tr>
<tr>
<td>Figure 3-5:</td>
<td>Disjoint classes by different criteria</td>
<td>18</td>
</tr>
<tr>
<td>Figure 3-6:</td>
<td>The activation of plane:0 by fly:0</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3-7:</td>
<td>Assorted hierarchical additions to context</td>
<td>20</td>
</tr>
<tr>
<td>Figure 3-8:</td>
<td>Simple role relation</td>
<td>21</td>
</tr>
<tr>
<td>Figure 3-9:</td>
<td>The organization of system partitions</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3-10:</td>
<td>Additions to context as a function of workspace and recency</td>
<td>24</td>
</tr>
<tr>
<td>Figure 3-11:</td>
<td>Lexical connections to concepts. All definitions from Webster's New Collegiate Dictionary [WNCD 74].</td>
<td>27</td>
</tr>
<tr>
<td>Figure 4-1:</td>
<td>Sample computed references</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4-2:</td>
<td>Reference to an implicitly present concept</td>
<td>32</td>
</tr>
<tr>
<td>Figure 4-3:</td>
<td>Lexical ambiguity reduced by context</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4-4:</td>
<td>Reference to a subquantity of an explicit concept</td>
<td>34</td>
</tr>
<tr>
<td>Figure 4-5:</td>
<td>A more complex reference to an implicit concept</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4-6:</td>
<td>Discourse structure after one sentence</td>
<td>37</td>
</tr>
<tr>
<td>Figure 5-1:</td>
<td>Sample translation to lexical predicates</td>
<td>39</td>
</tr>
<tr>
<td>Figure 5-2:</td>
<td>The reference process</td>
<td>39</td>
</tr>
<tr>
<td>Figure 5-3:</td>
<td>Disjointness restriction</td>
<td>42</td>
</tr>
<tr>
<td>Figure 5-4:</td>
<td>Descent restrictions</td>
<td>42</td>
</tr>
<tr>
<td>Figure 5-5:</td>
<td>Case 1: SA is in context. <em>A</em> instantiates A1 on SA1.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 5-6:</td>
<td>Case 2: SB in context, SB2 already instantiated. <em>A</em> specializes SB2 to an SA and instantiates an A on SB2.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 5-7:</td>
<td>Case 3: C in context, C3 instantiated. <em>A</em> instantiates an SA and creates a ROLE link from C3 to A.</td>
<td>45</td>
</tr>
<tr>
<td>Figure 5-8:</td>
<td>Relations between sense, usage, and referent</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5-9:</td>
<td>Preference ordering of candidate referents</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5-10:</td>
<td>Additional links added to an instantiated concept</td>
<td>49</td>
</tr>
<tr>
<td>Figure 5-11:</td>
<td>Noun phrase modification</td>
<td>50</td>
</tr>
<tr>
<td>Figure 5-12:</td>
<td>Noun phrase constituent referent search</td>
<td>50</td>
</tr>
<tr>
<td>Figure 6-1:</td>
<td>Sample texts</td>
<td>52</td>
</tr>
<tr>
<td>Figure 6-2:</td>
<td>Summary of results by reference category. Failures are in parentheses.</td>
<td>52</td>
</tr>
<tr>
<td>Figure 6-3:</td>
<td>Reduced texts</td>
<td>53</td>
</tr>
<tr>
<td>Figure 6-4:</td>
<td>A trading voyage initiation</td>
<td>54</td>
</tr>
<tr>
<td>Figure 6-5:</td>
<td>The planning schema</td>
<td>54</td>
</tr>
<tr>
<td>Figure 6-6:</td>
<td>A simple story with preconditions enforced via activation</td>
<td>55</td>
</tr>
<tr>
<td>Figure 6-7:</td>
<td>The want and give schemas with activation</td>
<td>56</td>
</tr>
<tr>
<td>Figure 6-8:</td>
<td>A simple story with activation</td>
<td>57</td>
</tr>
<tr>
<td>Figure 6-9:</td>
<td>Some effects of tense on reference</td>
<td>57</td>
</tr>
<tr>
<td>Figure 6-10:</td>
<td>Time magazine story beginning</td>
<td>58</td>
</tr>
<tr>
<td>Figure 6-11:</td>
<td>Conceptual sense for <em>rest</em></td>
<td>59</td>
</tr>
<tr>
<td>Figure 7-1:</td>
<td>The integration of &quot;murderer&quot; with die:11</td>
<td>65</td>
</tr>
<tr>
<td>Figure 7-2:</td>
<td>Two representations of schema results</td>
<td>66</td>
</tr>
</tbody>
</table>
Abstract

This report extends the concept of reference resolution in a discourse context to cover a broad range of connective inference required for text understanding. Access to all conceptual relations is restricted or facilitated by the context established by preceding text. This contextual filter greatly simplifies the establishment of connections between the surface text and previously instantiated discourse representation.

The reference procedure requires a taxonomically organized knowledge base of structured concepts, in the sense of frames and scripts. The procedure selects lexical senses and generates reference candidates, which may be either explicit or implicit in the discourse context. These are matched against constraints imposed by the surface text and a conceptual representation is constructed and integrated with the accumulated discourse structure.
1. Introduction

A great deal of the connectivity of texts is of a mundane sort in the sense that the relations between discourse objects are instances of well understood concepts. A scaffolding derived from this previously organized knowledge must be constructed before genuinely new information can be understood. In this report we extend the concept of reference resolution in a discourse context to cover a broad range of connective inference required for text understanding. A critical component of this procedure is that access to all conceptual relations is restricted or facilitated by the context established by preceding text. This contextual filter greatly simplifies the establishment of connections between the surface text and elements of the previously instantiated discourse representation. The reference procedure selects lexical senses and generates reference candidates, which may be either explicit or implicit in the discourse context. These are matched against constraints imposed by the surface text and a conceptual representation is constructed and integrated with the accumulated discourse structure. This construction is driven by the attempt to tie successive fragments of surface text into the growing discourse representation.

The procedures that we describe are limited to the utilization of conceptual knowledge, the semantic and world knowledge that pertains to the subject of the discourse. Thus they do not have access to syntactic knowledge or to the *textual syntax* that deals in meta rules for text processing. From the viewpoint of traditional syntax we are primarily in the realm of semantics and pragmatics. From the viewpoint of text-grammarians we are exploring the inferential black box that is necessary to determine the applicability of rules of text formation and intertextuality [see de Beaugrande 80].

The construction of a coherent discourse structure is fundamentally dependent upon an ability to recognize that a current phrase is the textual realization of some concept present in the discourse structure computed so far. That concept may be either explicitly or implicitly present in the discourse structure. We call this mapping from text to objects in the discourse representation reference. The term seems apt despite connotational clashes with its use in formal logics. [Clark 77], [Johnson-Laird 77] and [Lockman 80] use the term similarly. See section 7.) Reference is then the process of mapping from a referring textual expression, the reference, to a conceptual object, the referent, in the discourse representation. The kinds of connections established in the course of reference computation form a large subset of those Clark [Clark 77] describes as *bridges*. These include anaphora in general and a wide class of schema filling or *expectation* satisfying operations. 1-2 provides some examples.

1-2. NBC agreed to pay $2 million to settle a sex discrimination suit brought by women employees. As acknowledgement that the network has practiced discrimination, $1.6 million in raises and back pay will go to former and present employees. [Time 77]

There are a number of different sorts of reference in this sample. The simplest are the explicit references. *Former and present employees* refers back to the concept created for *women employees*.1 *Employee* provides the lexical connection to the same generic concept in both cases. The first occurrence results in the creation of an instance of the generic concept. The second occurrence causes the initial instantiation to be expanded due to the modification by *former and present* and by virtue of the imbedding of the second occurrence in the second sentence. A similar reference link is present between *discrimination* in the second sentence and *sex discrimination* in the first.

*The network* in the second sentence of 1-2 refers to the concept created for *NBC*. In order that this reference be easily computable, the fact that NBC is a broadcasting network must be readily available. The conceptual structure created for *NBC* is referenced in the second instance by a less specific lexical item than was used to establish it initially. This less specific lexical item is only appropriate because of the context that has been created as a function of the previous discourse.

The analysis of *will go* in the second sentence of 1-2 can proceed in two ways. The standard sort of analysis would include a lexical entry for *go* that was a *transfer*. Demons in the argument slots would

1In discussing examples of reference we will frequently use a shorthand of the form "*text1* refers to *text2*" when the proper description is "*text1* refers to the previously created representation of *text2*".
fire when sufficient information was provided to establish a more specific interpretation. In this case the transfer of money would suggest a paying. An alternative kind of facilitation is present in this text and is the one that we have chosen to use. The previously established presence of *pay* in context in the first sentence suggests an interpretation of *will go*. When looking for a referent for *go*, its taxonomic descendants are examined. We insure that only those in context are visible. There is only one, the previously instantiated paying. By virtue of its heightened accessibility this sub-sense is the preferred interpretation. It is found to be a plausible match due to its monetary theme. Both procedural attachment and contextual filtering are capable of generating incorrect analyses. Our interpretation of the sense of *will go* results from the use of the context established by the preceding text as an initial focus for reference resolution. Lexical disambiguation is a part of this process. We prefer the use of this general mechanism to the implementation of another device.

The first sentence of 1-2 contains an important example of an implicit reference. How is it that we know that the "women employees" are in fact employees of NBC? There is no overt statement to this effect. NBC is known to be a corporation and corporations employ employees. It is thus a plausible inference, within the restricted context of the growing discourse, that the employees are in fact employees of NBC. It is possible that there are no previous instances of employees as parts of the existing NBC concept, but we create one by virtue of inheritance across the taxonomic connection to corporation. Implicit reference of this sort motivates a great deal of the plausible connective inferencing that takes place over texts.

*Pay* in the first sentence of 1-2 is another example of a reference to an implicit concept. In a settled suit there is a compensation agreed upon between the plaintiff and defendant. *Pay* in 1-2 refers to the implied compensation of the settled suit. In the knowledge base applied to this text, paying is a more specific concept than compensate. Our reference algorithm utilizes the contextually organized data base to compute this equivalence of an implicit compensation and a textually presented paying.

The preceding examples indicate that reference as we define it cuts across a number of categories of textual connection that have tended to be considered in isolation. Included are lexical sense selection, anaphora including definite NP and VP reference, the slot filling associated with frames, and several kinds of connective inferencing. The referents that we compute are objects that are either explicitly a part of the discourse representation or implicit in the discourse by virtue of being taxonomic superiors of explicit concepts or subparts of these superiors. The path from a lexical sense to one of these objects is restricted to following taxonomic links or binding links between frame slot descriptions.

One example of the class of reference we cannot handle can be seen in 1-3.

**1-3.** Thousands of people turned out to see the first men to set foot on the moon. The heroes were given a wild welcome.

Our system would be unable to decide between *thousands of people* and *the first men on the moon* as referents of *the heroes* because of the added inferential complexity involved in recognizing that being the first to the moon is somehow heroic.

We describe a procedure for computing a large class of mundane references. Input to the procedure is in the form of lexical predicates (see section 5.1). The procedure requires certain features of the representation and an explicit realization of a context as a function of the preceding discourse. It processes NP and clausal references in the same fashion. It differs from other proposals in its non-inferential flavor.\(^2\) Connective inferences tend to result from successful reference resolution, rather than driving the reference computation. It is the organization of the rules that supports inference, not the semantic content of the rules. The organization of concepts in our knowledge base determines which ones are accessible through context and which are not. The available concepts are the prime referent candidates. The fact that an accessible concept is (for example) the RESULT argument of a previously instantiated concept is of some concern to the matching algorithm, as it might violate some restriction on the reference, but this link contributes nothing to the initial generation of candidates.

\(^2\)By non-inferential we mean that modes ponens or a quest for true propositions is not what drives the reference search. Of course any representation of world knowledge contains data analogous to logical implications.
1.1 Outline of remainder

The remainder of this report has a hierarchical structure. Chapter 2 provides an overview of results. The later chapters generally expand upon information briefly described in chapter 2, except for chapter 6, which is a complete description of the results of the computational experiments. These two chapters should provide the reader a reasonable idea of what we have accomplished.

Representation issues are covered in chapter 3. A rough continuum for representation systems is presented and the features required of a representation designed to facilitate reference resolution are described. This chapter is relatively lengthy because aspects of the representation are inseparable from the reference algorithm.

Chapter 3 also describes the four major system partitions and their use. UniV contains all non-lexical information. Lexicon provides the connections from lexical items into uniV. The representation of the discourse is constructed in workspace. Context, which both constrains and facilitates the integration of successive pieces of the discourse, is a function of the representations in workspace. It includes the top level concepts of workspace and a small set of elements of uniV.

Chapter 4 presents a detailed set of reference computations taken from the most involved of our sample texts. It should provide a feel for the complexity of the possible relations between terms and the significant effects on discourse structure that can be produced in the course of reference resolution.

The reference algorithm is described in detail in chapter 5. Conceptually, the reference process returns a list of candidate references sorted according to the various preference rules. Functionally, the reference algorithm generates references in an order dictated by the preference rules. The matching required in order to test the complete suitability of a candidate is presented, followed by a description of the possible structural modifications dictated by a successful reference match. This chapter also covers the fine grained manipulation of context for knowledge integration within phrases, as opposed to between phrases and the discourse representation.

Chapter 6 presents the complete set of reference examples from our computational experiments, explains those that failed and discusses what is required to extend our methods to cover them.

Chapter 7 relates our work to that of other researchers.

The final chapter discusses directions for future work and reviews our findings.
2. Overview

We categorize different referents according to three dimensions, all of which are dependent upon a well defined context. In this chapter we will first discuss context, then the three aspects of reference resolution, followed by a description of the effects of these aspects on the constructed result in the discourse representation.

2.1 Context

That the context of an utterance must be the fundamental external constraint on its interpretation seems tautologous. In order that this be the case, context, the representation system and the processing primitives must be firmly integrated. Schema based representations ([Brachman 78], [Hendrix 78], [Minsky 75], [Schank 77], and numerous others) reflect this need to integrate representation and context, though their justification is the more general one, that structuring is indispensable for the application of knowledge to intelligent tasks.

We use the term schema to refer to any structured conceptual object. A script [Schank 77] is a schema dealing primarily with the set of events that occur as part of a stereotypical activity. A frame [Minsky 75] is a schema that provides an expanded description of a class of objects, including relations between the object's parts and pointers to other schemas. Schemas are generalized diagrams of events and objects that have been related by contexts of experience. Initially, a schema can be a single episode, a saved context. Generalization from experience is still a mysterious process but it seems nonetheless clear that a primary rationale behind the aggregation of concepts into schemas must be their aggregation in contexts of experience.

Context is a set of currently active schemas. This includes those that we are constructing in order to represent the text and their conceptual ancestors. Thus if Ralph's car is in context then it will typically be the case that car:0 and vehicle will also be in context.

One of the truisms of AI states roughly that "anything will work in a microworld." The computation of a context as a function of the discourse encountered to date reduces the preferred referential search space to a microworld appropriate to the text.

Reference search proceeds from a fragment of input text back into the discourse representation. Our use of context as the preliminary filter on this search is critical. Without such a feature the potential set of paths to be traversed is overwhelming. (The alternative direction for search, from discourse representation to input text, seems a counterintuitive characterization, though Robinson [Robinson 80] presents a plausible application of this approach in a restricted domain. See section 7.1.2.) This pivotal significance of context is reflected in our implementation. All functions that access relations between concepts are limited by the current value of context. Information in context is more accessible. Some information may not be available at all if the necessary context has not been established.

In addition to this gross use, finer grained context switching provides a method for the analysis of possessives, nominal compounds, and adjective-noun combinations. These involve reference search within a narrow context that is a function of the dominant element of the phrase. In the case of noun-noun pairs, control is complicated by the indeterminacy of the contextually dominant item. Nonetheless, this approach, in which the relation computed between concepts is directly a function of their representation and, within a more restricted context, proceeds identically to contextual reference resolution, seems preferable to the use of a set of patterns specialized to nominal compound interpretation. (See section 5.6.)

2.2 Aspects of reference

Our reference algorithm suggests three critical aspects of reference. We distinguish explicit referents

---

3 Remember that we are ignoring syntactic effects. Many of the restrictions that would be applied in a complete system to map from surface syntax to conceptual representation are therefore slighted in our description.
from implicit ones. An explicit referent is one that has been instantiated in response to previous textual interpretation. It is a constructed piece of the discourse representation. An implicit referent is a contextually accessible portion of the knowledge associated with those concepts that have been explicitly instantiated. Secondly, we characterize the mapping from lexical item to concept according to its specificity. A lexical item may be used very precisely with respect to the category of its referent. That is, the immediate superclass of the referent may be one sense of the lexical item. The lexical item may also be more or less specific by pointing to a concept further removed from the referent in the taxonomy of concepts. For example, a previously instantiated instance of the concept dog0 may be referenced by any of the lexical items "dog", "animal", or "collie". Varying degrees of specificity in lexical expression are a fundamental part of written text. The third dimension of reference concerns the initial filter on the interpretation of input text. Some senses of a surface lexical item may be already present in context. These senses are preferred to senses that are not in context.

The combination of these three aspects determines the preference ordering of reference candidates. In addition, the various combinations of the first two determine the basic structure building operations to be applied to instantiate the input text.

Lexical items of varying degrees of specificity can be used to reference both explicitly and implicitly present concepts. Implicit references are a major component in the successful construction of a discourse structure. That lexical terms of varying degrees of precision with respect to the target referent can access both explicit and implicit concepts provides the broad scope of our reference algorithm.

2.2.1 Explicit vs. Implicit reference

A great deal of inferential leverage is acquired if the notion of "implicitly present in context" can be realized straightforwardly. Explicit concepts are those that have been instantiated due to previous input. Implicit concepts are reasonable targets of reference search because the larger concepts that they are defined in terms of have been added to context by virtue of explicit instantiation.

Reference to an implicit data object corresponds to expectation filling. In our system this is not accomplished by creating and checking an overt list of expectations but by making portions of world knowledge differentially accessible by virtue of knowledge organization. This distinction is perhaps more conceptual than actual. Context provides an expectation mechanism. The expectations dependent upon a particular concept are not computed dynamically but are effectively prestored by virtue of the organization of knowledge. There is a certain dynamism involved simply because concepts are combined to form contexts in new permutations. Any additional difference in viewpoint is due to the way in which these "expectations" are accessed. We begin at a conceptual representation suggested by surface lexical items and basically ask "Are there any of these in context?" The answer is found by determining whether the concept participates in any relations which have been made accessible due to context.

Example implicit references include the various frame and script based expectations.

2-2. John decided to sell his car. The brakes are shot.

2-3. Ralph went to a restaurant for lunch. As he was paying his bill, he tripped the waiter.

The definite NP, the brakes in 2-2 refers to a part of John's car which is implicitly present in context because world knowledge about cars has been brought into context due to the creation of an instance of one in the discourse representation. In 2-3 we have references to both a standard actor and a standard sub-event in the restaurant scenario. Because we know John is eating in a restaurant, we take advantage of the associated schemas which contains objects considered implicitly as part of context.

One problem avoided by the use of implicit reference as we describe it is the tendency to over-

\footnote{Many readers will stumble over "tripped", first interpreting it as "tipped". This is an example of the ubiquitous and powerful effects of context. An elaborate representational structure is priming the phonological (graphemic?) system.}
2-4. Jack walked over to the phone. He had to talk to Bill.

In our system, the instantiation of TELEPHONE would bring TELEPHONING into context. That JACK-1 is AT an object which is an element of the TELEPHONE schema would be asserted but no assumption would be made regarding Jack’s future behavior. *Talk* in the second sentence is integrated easily because it is an implicit element of the TELEPHONE schema. It is the *had* of the second sentence that indicates the *walking* was designed to satisfy the telephoning precondition.

2.2.2 Specificity of reference

The second dimension of reference has to do with the mapping between surface lexical items and target referents. We define three levels of specificity that correspond to the precision with which the surface lexical item characterizes the referent.

An index term provides a direct connection from the lexicon to the target referent or more typically to the class of which the target is an immediate instance. We will call this class the index class. In 2-5 *car* provides a pointer to the concept car:0^5 which

2-5. John sold his car and his bicycle last week. Mary said that the car was about to fall apart.

has an instance in context created in response to the first sentence. Note that an index term does not necessarily provide an unambiguous reference. Multiple instances of a referenced class are common. Specificity refers to the mapping between the lexical item and the general concept, not to the mapping between the lexical item and instances of the concept, of which there may be many.

2-6. John tested the drug on two rats before presenting his results. During the question and answer portion of his presentation the significance of his test results was vigorously disputed.

2-6 is more complex and illustrates cross syntactic reference. The same degree of specificity is involved, however. The nominal presentation references presenting and test refers to tested. Brachman [Brachman 78] contains a lengthy conceptual analysis of nominals, focusing on precise mechanisms required to represent various classes. We have found it advantageous to treat the conceptual representation of such syntactically different forms identically, leaving the differences as further research. We are primarily concerned with recognizing the referential force of text and the contextual priming that the first occurrence of a concept generates despite its syntactic form. The most important point to note in connection with 2-6 is that failure to make the connections between these syntactically different phrases is a failure to understand the text.

A specializing term indexes a conceptual class that is more specific than that of the target reference. It adds information.

2-7. John went to Europe last summer. Before he sailed he sold his car.

^5We will use the convention of associating unique names with concepts. These names will be printed in bold face with suffixed numbers.
2-8. Harry saw an animal in the bushes. When he approached, the tiger pounced.

It seems clear that one of the tasks required of a language processing system that is to be applied to 2-7 is to recognize that John's going to Europe last summer was a sailing event. Correct resolution of this reference results in an updating of the previously created proposition. Thus, a discourse outline like GO(JOHN,EUROPE) is transformed into BEFORE(SAIL(JOHN,EUROPE), SELL(JOHN,CAR)) where the GO has been subsumed by the SAIL. Similarly, 2-8 should result in a discourse representation in which the *animal in the bushes* is a *tiger*, not simply an unelaborated animal. This can be seen by judging the appropriateness of responses to the question "What did Harry see in the bushes?" The only reason to respond with *an animal* rather than *a tiger* is if one assumes the question was to be interpreted as "What did Harry think he saw in the bushes?" or "From Harry's point of view, what did he see in the bushes."

Stenning [Stenning 78] has ruled out anaphors in which the referring term is more specific than the referent. We include them because of the preceding examples and because of their utility in implicit reference. Examples of explicit reference like 2-8 seem to be acceptable because the two sentences are from different perspectives. In the first we are told what John perceived. In the second the narrator is describing what was actually present. This later elaboration might contradict John's perception, instead of specializing it. The perceived animal in the bushes might turn out to be a statue or John's friend wearing a gorilla suit. These are cases that we are not prepared to deal with.

Our primary use of specializing terms is in those cases in which they refer to an implicit element of the context. They allow schema slots with very general conceptual class restrictions to be recognized as filled by more specific instances of the basic slot descriptor. For example in 2-9, *catamaran* is the vehicle used in

2-9. Bob went sailing on Lake Austin today. His catamaran performed better than he had expected.

the sailing. The sail schema has a slot for a sailboat which is specialized in the instantiated schema to a catamaran, which is a more restricted subclass of sailboat.

A generalizing term provides a pointer to a less specific class than that of the target reference. A term for a more general class is used to refer to an instance of a subclass. Siderer [Sidner 79] has used the term lexical generalization to label this phenomenon. 2-10 and 2-11 are the inverse cases of 2-7 and 2-8.

2-10. John sailed to Europe last summer. Before he went he sold his car.

2-11. Harry saw a tiger in the bushes. When he approached, the animal pounced.

2-12. Bob's dog is incredibly obnoxious. The boy just doesn't know how to control the fool animal.

In 2-12 *animal* leads us to the concept representing Bob's dog by a path from animal:0 to dog:0 to bob's_dog:12.

Notice that "the boy" in 2-12 may be an index term or a specializing term depending upon the representation previously created for *Bob*. Names are a difficult class of lexical object. We are aware of their use to label particular individuals (thus their consideration as rigid designators in logical systems). In the case of the name *Bob* two reference results are possible. A known instance of a person named

---

6Depending upon knowledge organization or further context *sailed* might be interpreted as the departure involved in the first of a series of goings that make up traveling. In this case, *Europe* as the object of *to* would have specialized go to travel. Context exerts powerful intra-sentential as well as inter-sentential effects.
"Bob" may be retrieved from world knowledge and judged an appropriate match given context. This case enriches the interpretive context by making available the world knowledge previously associated with some particular individual. "Boy" in this case would likely be an index term because if we know much at all about the denoted individual we have already set up a representation that explicitly includes his "boyness". If on the other hand we create a new instance of a male person named "Bob", then "boy" specializes male-person:23 and adds information by restricting properties of the individual, for example the possible range of his ages, and by priming additional concepts for reference. The existence of these two cases is simply a reflection of the fact that different individuals have different conceptual structures organized in varying ways. As a result, one person's trivial reference is another's serious problem solving task.

The distinction between specializing and generalizing referents is suggested briefly in Robinson's [Robinson 81] speculation regarding bottom-up verb referent search in task oriented dialogues. See section 7.1.2.

Specializing and generalizing presuppose hierarchical relations between concepts. These may be explicit relations (ISA and its brethren) or derived, perhaps by virtue of primitive decomposition. We prefer explicit relations. Constant decomposition obfuscates. Concepts are describable at varying levels of detail. Lexical items are chosen with a particular level in mind. While a representation should provide for the possibility of reducing a concept to a lower level of detail, there seems little reason to insist that this always be the case. If such a decompositional approach is taken additional layers of representation must be added in the form of structured hierarchical concepts. (See [Lehnert 79] and [Bobrow 79] for a debate on the merits of decompositional approaches.)

It is important that there be many paths to explicit and implicit elements of context provided by specializing and generalizing terms. Language is marvelously fluid in use, but without a contextual filtering similar to what we propose the plethora of potential usages of lexical items would be difficult to reconcile with the precision with which a reader selects the appropriate one.

2.2.3 Lexical sense in context

The third aspect of reference processing is the initial filter that results from the contextual restrictions on the search for a set of likely senses for a lexical item. Senses are just concepts. Some concepts are in context due to the structure of the discourse. Given a lexical item with multiple senses, one of which is in context, it makes sense to explore first the sense in context. This is what our algorithm does.

2.2.4 Preference ordering of candidate references

Given the varied paths to referents, a discourse representation will frequently provide multiple possibilities for a single reference. One requirement we have placed on the reference computation is that it generate a preference ordering of candidates. Lexical senses in context are preferred to those not. Explicitly present concepts are preferred to implicitly present concepts, with totally new instances as the default. Within the first two of these categories, subordinate instances of the index class are preferred to superordinate instances. A subordinate instance is a referent derived from a generalizing term. The referent is lower in the conceptual taxonomy than the concept labeled by the referring term. A superordinate instance is one derived from a specializing term. The referent is above the concept labeled by the referring term.

We view the reference algorithm as a generator that suggests candidates in preference order. As implemented it in fact computes the preferred possibilities first, suspends itself, and returns the initial set of potential referents. If all of these are rejected it can resume and continue to produce further possibilities. This reflects our intuitive feel for the reference process. Some might argue that this is simply an implementation detail, that retrieving all possible referent candidates and then sorting them into preference order would be equivalent. Given the current state of our implementation this is true. But we believe it more appropriate to view conceptual networks as active processes that cascade chains of activation and inhibition. In this situation spurious activations would occur if all possible candidates were ever considered equally plausible before some filtering took place.
2.3 The effects of reference resolution

Reference resolution is fundamental to the construction of a coherent discourse representation. There are several basic structuring operations that may result from a successful referent search.

Perhaps the simplest structural effect is achieved by tracing the mapping from an index term to an instance already present in the discourse representation. The index term does not add information in and of itself, since it is referring to the referent using a label for a conceptual class of which the referent is already known to be a member. However, the surface term may have arguments or modifiers that add information to the referent. Or it may itself be an argument of a new proposition. In this case the referent assists in knitting together successive propositions of the text of which it is a shared subpart. 2-13 would be tentatively judged connected because the two sentences share an argument. This sort of connection has traditionally been considered a minimum (but not sufficient) requirement for a coherent text [Bellert 70, Grimes 75, Hobbs 79].

2-13. John bought a dog and a cat. The cat was Siamese.

Specialization adds information to the discourse structure by further restricting the class of which a concept is a member. Typically this makes available more detailed inferential material for the concept.

Successful reference to an implicit concept results in its instantiation within a matrix of connections to other previously created instances of concepts. This integration is one of the operations critical to the construction of a connected discourse representation. Having settled on a referent we examine the schemas of which it is part to see if its prototype is directly related to other concepts instantiated in context. Any such relations are then replicated between the concept instances. This is what we mean when we state that connective inferences result from correct reference resolution. For example in 2-14 "pay" will be found implicitly on the concept created for "bet".


After it is instantiated it will be linked to the "lose" instance by a RESULT link because this link exists between the prototypical lose and pay on the bet schema.

In order to instantiate an imbedded new concept it is necessary to find or create an instance of the concept that it is defined on. It is also possible that a newly created instance inherits activation links (see section 3.4.1.4). In either case the surrounding schema or activated concepts initiate a reference search in the same way that textual input does, simply bypassing the lexical interpretation stage. This may result in the specialization of previously instantiated schemas which in turn may alter the way in which the earlier concepts are viewed, given that their matrix of connections must now be interpreted in the context provided by the new schema.

2.4 Summary

The ideas presented in this chapter are summarized in the following outline.

1. Context restricts access to conceptual relations according to the organization of conceptual knowledge and is a function of the current discourse structure.

2. Three aspects of reference determine the gross order of generation of reference candidates.
   a. Explicitly present referents are those already instantiated in the discourse representation.
      Implicit referents are those that are present in context but not yet instantiated. They are present in context because the concept of which they are a subpart has a descendant

---

7 An imbedded concept is one that is defined only in the context of another.
instantiated in the discourse representation. Explicit referents are preferred to implicit referents.

b. A lexical item may point to a referent with varying degrees of specificity. It may label the concept of which the referent is an immediate instance, in which case it is an index term. It may label a concept which is a more specific subclass of the existing referent, in which case it is a specializing term. Finally, it may label a concept which is more general than the immediate conceptual class of the referent, in which case it is a generalizing term. Generalizing and index terms are preferred to specializing terms.

c. A word may have multiple senses. Senses in context are preferred interpretations of surface lexical items.

3. Resolution of input lexical predicates may have a number of different effects on the discourse representation. Additional modification may be added to a concept. An existing concept may tie together successive input propositions because it is an argument to both. A specializing reference more precisely specifies the conceptual class of which an instantiated concept is an instance. Reference to implicit concepts knits together the discourse by filling in expected subparts of larger concepts. The integration of implicit concepts results in the creation of additional links between concepts based on their defining schemas.
3. Representation Issues

If context is to provide a preliminary filter on processing, our knowledge representations and processing primitives must facilitate this filtering. The constraints on representation that we begin with are that it provide for structured bundles of associated concepts (frames, scripts, plans, etc.) and that there be some taxonomic organization of concepts. The algorithms for reference resolution that we describe require these properties of the representation. We take no position on many questions of representation because so many of the problems of appropriate knowledge representations for natural language processing are unanswered or their purported solutions are debatable.

The following sections present a brief discussion of representation in general, a description of the representation system used, the procedural motivation behind its various aspects, and the four major divisions of our representation space. Limitations of our representation are discussed in 3.6.

3.1 Levels of representation

Theories of cognitive science are designed to explain the behavior of powerful visual, linguistic, and problem solving systems. There are a variety of legitimate approaches to this task. In vision, rigorous mathematical methods can be developed to describe how certain classes of information can be derived from two-dimensional, binocular arrays of intensity and frequency. For other aspects of intelligent behavior we are forced to use other methods of verification. The cognitive psychologist requires that a theory make predictions which can be tested. This task is complicated by the vast power and flexibility of human processing. The AI researcher may embody a theory in a running program to see if the desired behavior can in fact be mimicked by the processes and representation that compose the theory. This method of verification may depend on the scientist’s programming ability rather than the quality of the theory, and thus must be weighed against other considerations, i.e. the theory’s coverage and its simplicity and clarity.

Theories of cognitive science may be presented at various levels of abstraction. Some, such as a theory of belief or personality, must rely on assumptions about lower levels of cognitive processing. Theories of human information processing should in principle be decomposable down to the physiological level. Computer models branch off from this hierarchy of abstractions to actually implement one such layer on a machine (see figure 3-1).

![Diagram](3.1 Levels of representation in cognitive theories)

The transitions between levels should describe how the lower level of process and representation can be used to provide the abstractions of the next higher level. At the top of the tree is an instance of a theory. In general, this is what we have available for purposes of testing. An experimental subject is one such
instance, a program with fixed data another. A program has the combined advantage and disadvantage of being isolated from other components of a whole theory of human intelligence. Other aspects of information processing do not interfere with the model. At the same time, these other aspects might completely rule out the model’s existence within a larger framework, a fact that may go undetected in isolation.

Corresponding to the different layers of this taxonomy of theories is a taxonomy of representation, including four levels: the physical level, the program level, the formal level, and the instance level.

3.1.1 Physical level

The lowest level is the physical level: the hardware (silicon or synaptic) and associated instruction set. The physical machine on which a computer model is implemented typically has nothing to do with the physiological level of cognitive theories. Serious treatment of the physiological level of representation is beyond the scope of this paper. However, assumptions regarding the features that it provides may intrude into system design. For example, in a system employing tree search without parallelism it becomes necessary to provide an explanation for the operation of tree-pruning heuristics. Fahlman [Fahlman 78, Fahlman 80] has shown that certain aspects of intelligent information retrieval are conceptually simpler when considered implemented on special purpose hardware. This is not to be confused with vacuous appeals to parallelism as a panacea. Fahlman’s proposals are a carefully worked out integration of hardware and formal representations.

One important point to note regarding the physical level of representation is that the description of a computational engine of any sort is incomplete if it fails to cover both the primitive objects manipulated by the machine and the operations defined over them. Only in the union of the two can we adequately evaluate its usefulness. The same holds true at other levels.

3.1.2 Program level

Program representation deals in neutral data objects and their manipulation. This level of representation is only appropriate to the branch of the theory abstraction tree concerned with the implementation of a model as a computer program. Elements of this level include such diverse notions as integers, atoms, lists, arrays, records, nodes, arcs, graphs, etc. The typical set of operations over these objects includes methods for creating, altering, comparing, and otherwise manipulating them. They span a wide range. Many are definable in terms of others. They are the data types of computer science.

Most of the computational metaphors that we bring to cognitive science arise from experience with objects at this conceptual level. Sometimes programmatic descriptions are proposed as theories of representation at the formal level. Semantic nets without an accompanying description of the interpretation of links, as critiqued in [Woods 75], are a good example of this difficulty.

Part of the absorption in such descriptions is a function of the engineering aspects of cognitive science. Models that are embodied in programs need to function if we are to learn from them. Part of our collective experience with such models has been that implemented programs frequently become considerably less elegant than the formal theory which gave rise to them. The reality of computation is one test of a theory. Failure to compute may or may not invalidate a theory of cognition, but successful computation adds plausibility. Additionally, the precise generation of a working model forces a principled investigation of a theory’s implications.

3.1.3 Formal level

It is on the formal level that most representation systems focus. Formal theories correspond to the theory through theoryN levels of the abstraction tree in figure 3-1. The objects and procedures of this level are those provided by AI programming languages and representation theories. It is here that frames, scripts, and particular sets of representation primitives are elaborated into a representation vocabulary for concepts.
A formal system defines primitive conceptual units and operations over them in terms of the next lower level of abstraction. Computationally this definition is initially provided by expanding programatic units. A node in a graph may be defined to correspond to a concept. Different kinds of concepts corresponding to differently marked nodes may be defined. A partitioning of nodes may be given some interpretation in terms of the formal theory. Various different kinds of partitions might be defined to carry different interpretations.

The features of a formal representation that can be considered of significance are those whose definition extends to a description of their interpretation in terms of processing primitives. This is the important point in the evaluation of any representation scheme. It is not critical that particular classes of information be literally present in a representation, but it may be necessary that they be computable. (See section 3.2.)

3.1.4 Instances of representations

The high end of the continuum is the instance level. It is here that one might debate the correctness of instances of representations of particular concepts and their relation to the formal system. For example, one might ask what rules of rhetoric are designed to accommodate inherent biases in the formal system and what rules are culturally shared? Part of the difficulty involved here is in defining the shared kernel of a concept, distinguishing idiosyncratic processing from the norm.

3.2 Representation interpretation

No representation system is complete without a description of the primitive operations defined over it. In AI this is particularly obvious. Cognitive psychology, often forced to resort to reaction times to deduce what might be going on in our heads, must of necessity consider the various possible interpreters for the system.

Only in the procedural/declarative whole can we evaluate the adequacy of theories of natural language processing. One classic difficulty, for example, is the attempt to specify an adequate representation scheme in the absence of an interpreter. This has often led to a misperception of what it means for a representation to be adequate. An adequate representation is one that in conjunction with the primitive procedures defined over it supports the computation of specified classes of information. This information does not have to be literally present in the declarative portion of the representation. One concern has been that a system easily provide a general mechanism for the representation of arbitrarily quantified expressions of English [Hendrix 78, Montague 74, Shubert 78, Simmons 79a]. People have very powerful rules of quantification that may be easily and systematically misled [Johnson-Laird 77]. The fact that logicians and computer scientists can extract the multiple interpretations of "All men love a woman" implies that it must be possible for the rules of logical quantification to be represented given the rules of natural quantification that are built into the underlying representation methods that humans use. If we wished to analyze example texts that dealt with quantified predicate calculus we would need to develop representations for logical quantification in terms of the primitive representational and computational components of our theory.

Of necessity we have created a reasonably systematic representation but make no claims for its completeness. What is critical is the integration of representation and computation. The algorithms for reference resolution that we describe require certain properties of the representation.

3.3 Representation overview

Our programatic level of representation uses partitioned networks [Hendrix 78, Hendrix 75]. As

---

8 It is a predilection of the human processor to bring to bear a vast amount of knowledge to the interpretation of any input. It is difficult to resist assuming that the node MOVE3034 in a representation contains information similar to that of one interpretation of the word "move". It can often be seen that in some sense the information is present in a system, and the fact that the algorithms to extract and use it are unknown or ineffective is not obvious.
presented by Hendrix, partitioned nets have both programatic and formal properties. We have used their programatic properties and imposed a formal interpretation on these properties similar to their use in [Grosz 77a].

There are a number of reasons for our choice of partitioned nets. When treating nodes and partitions in nets as concepts, they suggest the immediately available pieces of information connected to a particular concept. The hierarchical relations that we have found critical for reference are tree structures that map very directly onto nets. Partitioning has provided a simple and convenient technique for both the structuring of concepts and the contextually sensitive manipulation of them. Nets reflect our intuitions regarding associativity. This is fundamentally a processing question. To quote Brachman [Brachman 78], "Associativity depends on how one has chosen to store objects and their relationships, not the relationships themselves." Because processing based on contextually sensitive data structures is at the heart of our reference algorithm, associativity is of fundamental importance.

In a given representation scheme an object acquires meaning by virtue of the application of its matrix of connections (explicit or implicit) to other objects. Data items capable of such meaningful interpretation will be called concepts. A structured concept is one that subsumes and organizes other concepts. We refer to these subsumed concepts as embedded concepts. In line with a frame-like view of the world we will sometimes refer to the instantiations of embedded concepts as "filled slots" of the parent concept. In fact virtually all concepts are structured or are subclasses of structured concepts. We use the term structured when we wish to emphasize the parts rather than the whole. We use nodes and partitions to implement concepts. Structured concepts are implemented by partitions. Links between nodes and partitions are not concepts. They are binary relations between concepts. System defined features are unary predicates over concepts.

Generic concepts are definitions, generalizations across experience. Extensional concepts are instances of generic concepts that are believed to have existed or occurred. We do not deal with hypothetical concepts which are concepts that are expected to exist or occur in the future.

A manifestation is a concept that is an instance in a context of some extensional instance in world knowledge. While the idea of having multiple nodes to represent the same entity may seem strange (though see [Bobrow 77], [Hayes 77], and [Brachman 78] which all use similar notions), it simplifies the task of maintaining contextually compartmentalized blocks of information about extensional entities. Manifestations also appear useful for reasoning about real objects in hypothetical contexts.

### 3.4 Required properties

The requirements placed on a representation system by our reference algorithm are outlined in figure 3-2. The mechanisms that satisfy these requirements are described in the following sections.

#### 3.4.1 Links

The objects in our formal representational space are roughly divided into trees of entities, states, events and acts. The basic reference algorithm treats them identically. The existence of these disjoint trees is simply a function of the particular conceptual data base that we have created.

The representation provides a facility for structured concepts containing role and subpart descriptions. Concepts are organized taxonomically. Part of the hierarchic relation between two structured concepts is an explicit description of the mapping between their subparts. For example if sail:1 is a subclass of go:3 we want to conveniently access the corresponding relation between their roles. Go:3 might have a role slot, vehicle:12. The corresponding role on sail:1, boat:10 is connected by a ROLE link to vehicle:12. ROLE links are described in section 3.4.1.3.

We use the same link name (ISA) to describe an assortment of hierarchic relations. It is possible to do so because concepts are of different types. The type of a concept is determined by a small set of features associated with it. These type distinctions are used by the procedures that run over the nets and when necessary precise distinctions can be made based on features of the related concepts. The reason the basic taxonomic link names are limited is that the primitive reference operation that we have defined treats the varied taxonomic relations identically.
1. A taxonomically organized database of concepts is necessary. There may be any number of disjoint taxonomic trees.
   a. A concept must always be able to determine its taxonomic superior, if it has one.
   b. In general, a concept should not be able to examine all of its taxonomic descendents.
   c. Taxonomic relations should connect the corresponding subparts of hierarchically related concepts.

2. Some frame-like partitioning of conceptual space is needed to support contextual structuring of conceptual space.

3. A limited set of features on concepts is required to distinguish individuals, mass concepts, and sets and extensional and generic concepts.

**Figure 3-2:** Representation requirements for contextual reference resolution

There are two classes of links, those known to the system and all others, which we call user links. The two classes of links are interpreted in fundamentally different ways. The system links include the limited set of hierarchical relations (ISA, EQUIV, and ROLE) and procedural links (*A*, NEQ) that are used by the reference algorithm. User links are basically role indicators in structured concepts. The names of these implementation specific relations are arbitrary and have meaning only by virtue of the use of identical link names in hierarchically related concepts. In this case they are assumed to name the same conceptual slot.

Except for the link to a concept's fundamental type, access to all links is a function of context. This is the major feature shared by system links and user links. This restriction is violated by type-links in the upward direction in order to satisfy constraint 1a of figure 3-2. It is reasonable that a concept always be able to determine its fundamental class. For example, from the concept sail:0 we can always determine that it is a kind of go:0, but from go:0 an enumeration of its various subtypes is restricted by context. In addition, one flavor of lexical link also violates this access restriction in that it is able to penetrate conceptual boundaries even if the penetrated concept is not in context. (These lexical activation links are drawn with a small circle at their intersection with the concept boundary. See section 3.4.1.4.)

The contextual restriction on accessing links works as follows. All concepts reside on some partition. The most all-encompassing of these, uni, resides on itself. From any given concept we can only access those links which are connected to concepts that are either themselves elements of a given context or which reside on structured concepts that are elements of that context. We have only explored the utility of this device in very restricted ways. In addition to its use as a filter in reference processing this simple yet specific statement of the relation between context and representation seems to offer convenient ways of dealing with variant views of an object (change the context and access to information pertinent to it may be enabled or disabled) and suggests techniques for the generation of text for different audiences by altering the context present during generation. This use of partitioning to control access is nearly identical to that described in [Grosz 77a], though more restrictive in that links between concepts do not reside on partitions. Their visibility is a function of the locations of the connected concepts.

### 3.4.1.1 ISA

ISA is the essential taxonomic link. It provides the basic paths for referent search. The features marking hierarchically related concepts permit us to interpret the ISA link in the various ways shown in figure 3-3. The repertoire of features is presented in section 3.4.4. Two sets of features are used in order to interpret precisely the ISA link when necessary. A concept may be marked as a set (SET) or individual (INDIV) or unmarked with respect to number. A concept may be marked extensional (EXT) or generic (GENERIC).
A ISA B

Figure 3-3: Different relations implied by the ISA link depending on the features of connected concepts.
(Note: CI = conceptual instance)

The most commonly used relations are subclass, conceptual instance, and individual instance. The subclass relation connects a more specific generic concept to a less specific one. Collie:0 is a subclass of dog:0 which is a subclass of animal:0. We do not make the KRL distinction between abstract, basic, and specialization categories [Bobrow 77] though an argument for such a distinction can be made on the basis of certain limitations on potential reference candidates (see section 5.2.2.1).

Instances of concepts are of two kinds. There are those that are considered to have occurred or to exist in fact, which we call extensional. The other sort of instance is a conceptual instance (CI). Consider our interpretation of the famous restaurant script. This structured concept contains subparts describing what restaurants look like, who works in them and what happens when people go to one. One of the important subevents described in this world is an eating event. We do not wish to duplicate all of our knowledge about eating at this point in the script, so we construct a conceptual instance of eat:0, the more general concept. A conceptual instance carries the modifications and additions to a concept that are specific to its organizing schema. The major distinction between a conceptual instance and a conceptual subclass is that a CI cannot be instantiated without a corresponding instance of its defining schema. A CI may be accessible through context because its schema is in context, even if the CI's immediate superclass is not in context. The various roles in the restaurant schema are also CIs. Waitperson:0 is a conceptual instance of person:0, an extremely rich concept. Only those items particularly salient for waitperson:0's role in the restaurant script are explicitly present in this schema. Numerous others are inferable on the basis of the conceptual instance relation to person:0.

The conceptual instance relation uses the ISA link to connect an element of a partition to its generic type. In figure 3-4 a pilot:12 is a person:0 who is the AG of a flying event. The concept pilot:12 is embedded on fly:2 and is therefore a conceptual instance of person:0, as opposed to a subclass of person:0.

An individual instance is a particular existing instance of some generic concept. Most instantiations are of this sort.

The subtypes of a concept may be disjoint classes. As a crude mechanism for handling these cases the concept is marked as having disjoint subclasses. This means that all of its subclasses are mutually exclusive. The EQUIV link allows us to create distinct disjoint subsetting categories for a single concept. Figure 3-5 shows how we might define the subclasses of animal:0 according to separate disjoint categorizations. Ideally the disjoint equivalent concepts would be introduced in contexts where their criterion for refinement were appropriate, animal:01 with diet and animal:02 with biological classification.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET/INDIV</td>
<td>EXT</td>
</tr>
<tr>
<td>Φ</td>
<td>generic</td>
</tr>
<tr>
<td>set</td>
<td>generic</td>
</tr>
<tr>
<td>indiv</td>
<td>generic</td>
</tr>
<tr>
<td>set</td>
<td>ext</td>
</tr>
<tr>
<td>indiv</td>
<td>ext</td>
</tr>
<tr>
<td>set</td>
<td>ext</td>
</tr>
<tr>
<td>indiv</td>
<td>ext</td>
</tr>
</tbody>
</table>
3.4.1.2 EQUIV, the manifestation link

The manifestation link is used to define two concepts as equivalent, providing two views of one concept in two different contexts. In the reference algorithm two EQUIV concepts are treated as if they were collapsed to one. The structure building operations use it to instantiate an extensional concept from world knowledge in the current context. The EQUIV link is useful because we permit a concept to be directly embedded in only a single larger concept and it may be necessary to view the concept in different ways for various purposes. Multiple views could be implemented relying strictly on partitioning, but EQUIV nodes seem to simplify processing.

3.4.1.3 ROLE, the subpart mapping link

The ROLE link provides the fundamental connection between the instantiated slots of concepts and the slot definition. It is both necessary for the straightforward implementation of reference resolution and useful for conceptual representation in general, supporting requirement 2 of figure 3-2. ROLE links provide a parallel taxonomic tree to ISA links that is sensitive to context. We have stated that upward ISA links are always accessible. This is not the case with ROLE links which are contextually sensitive. Suppose John, a male person, is a doctor and flies a plane as a hobby. Depending on circumstances reasonable references to John include "him", "the doctor", or "the pilot". However the last two possibilities are somehow less central to the concept John:3 than is his categorization as a male-
person:0. They are roles he plays in different contexts while he is always a male-person:0. These ROLE references to John will only succeed if their corresponding partitions are present in context.

Similar devices are used in [Brachman 78], [Hayes 77], and [Fahlman 78]. The reference algorithm treats ROLE trees and ISA trees in a nearly uniform manner.

3.4.1.4 The procedural links, *A* and NEQ

Encountering an outgoing activation (*A*) link results in an attempt to discover a referent for the concept pointed to. From a lexical node, this initiates search for a realization of the textual input. From a concept it forms an associative link which fires to connect up an instance of some associated concept, creating a new instance if necessary.

In some intimately related conceptual structures there is no obvious dominating concept to subsume the other. In these cases it was found necessary to tie concepts together with activation arcs. For example in chapter 5 we present the sample text:

Mary flew to Austin.
The door opened.
She exited.

The concept fly:0 has an instrument which is a plane:0 which has a part which is a door:0. In order for "the door" of the second sentence to refer successfully to the door of the plane that Mary flew in, the vehicle of the flying must be instantiated when fly:0 is. If it is not, when we seek a referent of "door" plane:0 will not be in context and the path leading to the door of the instr of the flying will not be discovered. The result of activation in this case can be seen in figure 3-6. When creating fly:11 the *A* link on

![Diagram](image)

Figure 3-6: The activation of plane:0 by fly:0

fly:0 causes the reference algorithm to be applied to the instr of fly:0. Since context was empty this merely resulted in a new instance of the instr, P:11, being added to fly:11. The critical effect was the addition of plane:0 to context so that the "door" referent could be seen to be the implicit part of P:11.

NEQ links are a relatively minor aspect of the representation that allow one to ensure that two conceptual nodes are never collapsed into one. For example, the representation for "choose X over Y" asserts that X and Y are NEQ. This link also appears in input propositions when it is used to mark two pronouns in a sentence as not referring to the same entity due to surface syntactic constraints.
3.4.2 Uniformity of reference across the different hierarchic relations.

The gross organization of our database is into a set of trees of structured concepts. If we consider the fundamental import of hierarchic relations to be the contextual inclusion of ancestral concepts and resulting inheritance of sets of structured concepts, then there seems to be a functional similarity between the hierarchies of classification, part/wholeness, location, and temporal inclusion. These various relations all create a context of implication. Take for example concept X in figure 3-7.

```
   B   C   D
   |   |   |
   ISA DURING ISA
   
   A
   |   |   |
   X
```

Figure 3-7: Assorted hierarchical additions to context

The addition of X to context should result in the addition of A, B, C, and D also. A is added because X is a part of it. A establishes a context for the interpretation of X. B and D are related by an ISA link to elements of context and C temporally includes X. These additions have two potential effects on further processing. A, C, or D may match something already present, further knitting together the overall representation. Additionally, the structural subparts of A, B, C, and D are now available as implicit reference candidates, thus augmenting the context of implication around X.

One additional hierarchic structure that can be traced during reference search is the binding path (ROLE taxonomy) between subparts of structured concepts. For example the lexical item *arrive* points to a subpart of the structured concept go:0 in figure 3-8. Sail:0 ISA go:0 and carries further information about the arrival that is specific to sailing. This subpart, dock:3, is lexically labeled *dock*. We directly relate the sail:0 arrival to the go:0 arrival using a ROLE link. As a result dock:3 can be referenced by either of the lexical items *arrive* or *dock*. ROLE trees provide the same opportunities for reference resolution as do ISA trees. They greatly increase the connectivity of our conceptual space and facilitate a wide range of lexical references to a single concept according to context.

3.4.3 Merging of word classes in concepts

That we seem relatively flexible in the face of differing word classes when computing reference relations was noted previously in section 2.2.2. This has implications for any knowledge representation scheme that we select. There must be a fundamental unity in the representation of verbs and nouns. Our (over) simplification treats both the noun and verb forms of *test* as lexical indices to the concept test:0. We are ignoring most of the fine semantic distinctions that can be made based on surface syntax. Brachman [Brachman 78] attempts to cover this difficult area, but ends up creating links for his conceptual structures that are syntactically derived and whose conceptual weight is unclear. For example, the representation of the factive noun *test* would be related to the representation for the verb *test* by a dfactive or nominalization link.

---

*We have implemented the temporal links AFTER and DURING, but have not experimented with their use to any degree.*
3.4.4 Features on concepts

A data base of world knowledge is the result of past integration of representations of events, states, and entities into a network that simplifies recognition and facilitates expectation and inference generation. Any such representation must allow a number of distinctions to be made.

Some of these distinctions seem procedurally more primitive than others. These primitive properties of concepts have been built into our system to satisfy requirement 3 of figure 3-2. Sets are distinguished from individuals. Extensional entities (objects believed to exist and events believed to have actually occurred) are distinguished from intensional or generic concepts. Negated concepts and mass and count concepts are also marked. Sets, generic vs. extensional distinctions, and negation were considered necessary originally. Features to mark mass concepts were added in the course of our experiments. Mass concepts had to be distinguished in order to generate proper subpart reference candidates.

Instances and conceptual instances may be sets. Instances may be elements or subsets of extensional sets. Our set notation is limited. For example a concept that is a set cannot be both a subset of a set and an element of some set of sets. This restriction was made for the sake of functional simplicity.

3.5 System partitions

There are four partitions established by the system. All lexical and conceptual data is stored in these system partitions. The first of these is univ. This is the universal space of conceptual information (world knowledge). Those concepts which do not reside on it directly are located on concepts which are in turn either directly or recursively located on univ. We construct the representation of a discourse on the workspace, a partition reserved specifically for that purpose. Context includes workspace, its elements and their conceptual ancestors, and concepts highlighted by the discourse. Concepts may be both elements of univ and in context. Because context is very fluid it is not implemented as a partition, though we display it in our figures as if it were. Lexicon is a partition whose elements correspond to print-images of words and phrases. Elements of the lexicon are linked by a set of activation relations ("A") to the concepts in univ that represent their senses.

3.9 depicts the relations among the system partitions. The links from the lexicon into univ are all
activation links. All other links are either ISA or ROLE links. The workspace contains L which has a component subpart J. These two together determine which elements of univ are included as elements of context. G, B', and C' are added to context because they are hierarchic superiors of L. H, K, and D' are added because they are superiors of J (assuming J is a newly created element of L. See section 3.5.3). B and C, and possibly A, may be used to refer to L. D may reference J. Both E and F may refer to an implicit instance of I on L.

3.5.1 World knowledge

The immediate elements of univ are readily available world knowledge. In our system, world knowledge includes all conceptual and experiential knowledge. The precise boundaries of world knowledge are difficult to specify. We explicitly reject distinctions such as semantic memory vs. episodic memory (For a discussion see [Johnson-Laird 77]). How does one first acquire a scenario, say for going to a restaurant? One way is actually to go. This episode is remembered. It can be used to understand stories about restaurants (though perhaps with limited effectiveness) and to guide behavior the next time one attends a restaurant. We don't visualize experiential information as disappearing, retrieval just becomes unattainable. Anderson [Anderson 76] and Minsky [Minsky 80] both make such irretrievability of experience an explicit assumption. It is our view that world knowledge is to a large extent the result of an accretive process which only distinguishes between experience and information absorbed in other ways by the differing richness of the memory traces. As it will be quite a while before cognitive science is ready to tackle the full rich range of memory, our knowledge representations are propositional structures and have no hooks for imagery or connotation that might support such a distinction.

Because experience is not stored differently or separately from generalization, the process of searching for referents may result in the recall of specific stored instances. That is, our reference procedures do not provide a complete mechanism for the retrieval of instances from memory is a function of the restricted nature of our processing. We have defined reference resolution as the connection of input to elements of context. When context contains no legitimate reference we default to the creation of a new instance or manifestation in context based on the structure of concepts and instances in univ. The construction of a manifestation is the result of the retrieval of a specific instance in memory.

3.5.2 Workspace

We assume that the discourse representation has a partitioned structure as does any structured concept, with the more central propositions containing the peripheral or inferable ones. This discourse partition we call the workspace. The structure of conceptual embedding built up in the workspace derives primarily from the structure of preexistent conceptual knowledge. This seems to be a weakness of our structure building operations. The dependencies created on the workspace should be a function of both this preexisting knowledge and the structure of the surface discourse. We currently rely heavily on the typical, as embodied in world knowledge, rather than the novel, as embodied in the input text.

At all times the structure in the workspace has a limited summary property [Simmons 79b]. We can generate a summary or paraphrase by beginning with those partitions that reside directly on the workspace and expanding them to varying depth as required. This is a limited summary property because we are not applying rules of text grammar to the input and as a result any summary generated will be more a function of the organization of world knowledge than the organization of the text itself.

3.5.3 Context

Context is a function of the workspace. Its role is to control access to the concepts in univ. It does so in two ways. First, those elements of univ that are also in context or that are elements of concepts in context are preferred candidates for use in reference computation. This reflects the simple expectation that the elements of a discourse will more likely be related to concepts connected to preceding portions of the discourse than to unrelated concepts. This is the minimum that a context can provide.

---

10. Immediate elements are located directly on univ. Because partitioning is recursive there may be embedded concepts that reside on partitions that reside on univ. These and those even more deeply embedded are not immediate elements of univ.
Figure 3-9: The organization of system partitions
Secondly, **context** performs an unlocking function, allowing access to concepts that cannot be reached otherwise. Such concepts reside on non-system partitions and are only accessible if the father partition is an element of context.

**Context** is dependent on the discourse to date. It is computed as a function of **workspace**, which contains the representation of the discourse. Those concepts that reside directly on the **workspace** are used in computing **context**. The structure of the discourse representation keeps **context** from becoming cluttered or unmanageably large because most of the instantiated concepts do not reside directly on **workspace** but are subparts of concepts that do. In addition, selected subparts of **workspace** elements are added also. Frequently it is the case that a text expands on an embedded concept or references its implicit subparts. We put all concepts referenced in the preceding sentence of surface text into **context** also. This provides an emphasis on the last sentence that is similar to the preferential treatment that Lockman and Klappholz [Lockman 80] accord it. We do this without blocking reference to other elements of discourse context, however.

Adding a concept to **context** causes its superclass and any concepts it is located on (if other than **univ** or **workspace**) to be added also.

```
| F |

\[\text{TISA}\]

| E th \rightarrow \rightarrow | X | ←- |

\[\text{ROLE}\]

| | T |

\[\text{univ}\]

| workspace |

\[\text{ISA}\]

| E' th \rightarrow \rightarrow | C' | ←- |

\[\text{ISA}\]

| B' | A' ag |

\[\text{ISA}\]

\[\text{ISA}\]

| B | A ag |

```

**Figure 3-10**: Additions to **context** as a function of **workspace** and recency

In the case of figure 3-10 suppose we had just created \((E' \text{ th } C')\).\(^{11}\) The only concepts in the figure not in **context** would be \(A'\) and \(A. B'\) and \(E'\) are at the top level of the **workspace** so they and their superclasses, \(B\) and \(E\), are added. \(E\) **ISA** \(F\) so \(F\) is added. \(E'\) **ISA** \(C'\) so \(C'\) is added. \(C'\) resides on \(B'\) so \(B'\) and \(B\) are added. Notice that many of these concepts may be added to **context** for more than one reason. This suggests various weighting schemes designed to further refine preference within context, but our experiments in this direction have been unfruitful.

\(^{11}\) In text we use a parenthesized notation for nets. The first element of the list is a concept whose outgoing links to other concepts are described by the successive pairs of elements of the remainder of the list.
It is important to note that concepts may be accessible as reference candidates because they are elements of context or because of the inclusion of their home partition in context. In figure 3-10 A is accessible because B is an element of context even though A itself is not an element of context.

More formally, **context** is a set of partitions. Consider the set S.

\[ S = \{\text{workspace}\} \cup \text{conceptset(workspace)} \cup \text{last-phrase-set} \]

S is the set of concepts in the discourse representation from which **context** is computed. **Last-phrase-set** is the set of concepts created or referenced during the construction of the representation of the most recent phrase, normally a sentence. **Conceptset** is a function that returns the set of concepts that reside directly on a particular structured concept.

\[ \text{context} = \bigcup_{x \in S} \text{superset}(x) \]

**Context** is the union of the supersets of all elements of S. The superset of a concept, composed of its hierarchical ancestors, is found by following ISA and ROLE links up to the roots of trees. This is necessary if **context** and structured inheritance are to work together.

Taxonomies have traditionally been intended to provide some structure to concepts, providing answers to questions like "Is a canary a bird?" It was also recognized that such a structure could support the inheritance of properties [Quillian 68]. For example, by virtue of being taxonomically subordinate to **bird**:0 it is possible to infer that an instance of the class **canary**:0 can fly, unless the inherited property is explicitly negated at the lower level. Similar processes are at work in reference.

The lexical item "bird" can refer to a warm blooded vertebrate, a clay pigeon, a person (*I met a queer bird yesterday.*), or a girl (chiefly British). When "bird" is used to refer to an existing instance of **canary**:0 in context we want to prefer the vertebrate sense of the word. The simplest way to do so is to have added the taxonomic superiors of **canary**:0 to context when we created its instantiation. As a result the vertebrate sense of "bird" will be in context and preferred. Similarly, when we use the lexical item "fly" and **bird**:0 in **context** the preferred usage/sense is the flying that a bird does. Because this relation between **bird**:0 and **fly**:0 might not be duplicated on **canary**:0, again it is necessary that **bird**:0 be added to **context** whenever **canary**:0 is.

We have taken a strictly binary view of **context**. A concept is either in or out. There is no weighting. This binary structure is at times inadequate. A preferred subset of context, the primary focus of [Grosz 77b] or the immediate focus of [Sidner 79] are needed sometimes to decide between competing reference possibilities, to guide the structure building needed for new schemas, and to support certain syntactic processes (ellipsis). See section 7.1.2.

### 3.5.4 Lexicon

The print images of words reside on a partition called the **lexicon**. They provide indices to concepts in **univ**, their senses, in three ways. A sense may be a top level concept in **univ**, in which case it is always accessible from the lexicon. This is where the senses of the great majority of everyday words lie. The second class includes those that are always available but which are intrinsically part of some larger schema. Finally there are senses that are only available if the appropriate context has been established.

Particularly obvious in the second class are senses of words identifying occupations and other idiosyncratic schema arguments.

3-2. Bob was a **waiter** last summer.

3-3. The **quarterback** was sacked with seconds to play.
3-4. I need to get my carburetor fixed.

3-5. Peter departed last week for Hamburger University.

In varying degrees the marked phrases in the above sentences invoke schemas in which they are involved. "Waiter* alone in 3-2 is sufficient to increase the reader's expectation of restaurant related concepts.12 In 3-3 "quarterback* invokes the football schema, which simplifies the comprehension of "sack*. Discussion of carburetors in 3-4 is impossible without the association of engine or automobile concepts. Actions may also be identified as basically a part of some other action. The "departing" in 3-5 is the first act in the execution of the schema associated with one sense of "go*.

There are two mechanisms for making these intrinsic associations in our knowledge structures. Concepts that are subparts of larger schemas and at the same time readily accessible from the lexicon are connected to the lexicon by links that can penetrate the enclosing partition. When these links are followed, the enclosing partition is temporarily added to context pending successful reference resolution against the embedded concept. Links of this sort are indicated in our figures by a small circle at the juncture of the link and the partition. We call the concepts pointed to in this way embedded, activating concepts.

For associated concepts for which no clear subsumption exists we use activation links, which force the addition of the associated concepts to context. "Hammer and nail* and "fly and plane* are examples of this sort. The reason for these two methods of activation is that we are using partitions to structure the discourse in addition to serving to implement contextual activation. Contextual activation is primarily a function of structure but in some cases it is necessary to add an activation link rather than distort the organization of knowledge.

The third sort of lexical sense does not even occur to a reader unless the appropriate context has been set up. This seems particularly clear in the case of obscure senses of common words (3-6, 3-7, 3-8).

3-6. John's boat needs new sails and the head needs a new door.

3-7. Pete Rose was going to sacrifice but ended up walking.

3-8. Sam is a Gemini. His house is ill-favored

Presumably one of the reasons that people cannot rattle off the various senses of words with the facility of a dictionary is that a significant number require some contextual priming to access.

A single lexical item may have all three classes of sense. For example "head* has a verb sense, head:3-2 in figure 3-11 which is in the first class, always available. This sense of "head* is not the most likely to come to mind given the word in isolation. Head:1-1 seems conceptually to belong as a subpart of the body:01 concept, but appears to be accessible at all times. It is in the second class, a sense that is a subpart of another and introduces the whole to context. While it was being considered as a candidate for reference head:1-1 and body:01 would both be added to context.

Head:1-12b is in the third class, a subpart which is inaccessible unless the enclosing partition is an element of context.

---

12 Of course this expectation may easily be violated. The frequent violation of such expectations is one reason we prefer to handle them using methods that involve low overhead.
1. head:3-2 = to point or proceed in a certain direction.

2. head:1-1 = the upper or anterior division of the body (as of a man or an insect) that contains the brain, the chief sense organs and the mouth.

3. head:1-12b = a ship's toilet.

**Figure 3-11:**
Lexical connections to concepts.
All definitions from Webster's New Collegiate Dictionary [WNCD 74].
3.6 Limitations of representation

The representation is incomplete. We have intentionally avoided as many of the difficult problems of representation as possible, hoping to make some progress on the aspects directly related to the reference algorithm.

There are no tested primitive representational devices to deal with time or causality. The representation of belief is not integrated. "I know John believes that Mary thinks I believe in elves. She doesn't but I do." We would like to approach belief representation as a special case of contextual manipulation. For example, in order to speak innocuously to someone whose beliefs you know to diverge from yours, an appropriate strategy might be to set the context for generation to include his belief set and exclude yours and then simply speak as you normally would. This is the same method one would apply in speaking to someone whose knowledge differed from yours. A mechanism would be required to deal with the fuzzy boundary between "know" and "believe".

We support multiple views of concepts in two ways. A concept may have multiple manifestations that focus on its different aspects. Or a concept may have links to various sets of propositions that are not available unless their partition of residence is in context. We have not explored the use of such multiple views in connection with reference.

We take a restrictive approach to ISA links. We do not allow multiple upward ISA links originating from a single concept. ROLE arcs provide some capability for multiple hierarchic superiors. We assume that a concept has a basic identity as a member of some class and less central inclusion in others.

Despite the ubiquity of metaphor [Lakoff 80] we have no devices to deal with it.

There are a number of function words for which we provide no good account. Anaphors like "one" and "most" seem more easily handled as functions applied to the discourse structure rather than as simple lexical pointers to conceptual structure. Adverbs like "too" seem to require similar representations. We have completely ignored the varying effects of the definite and indefinite articles, negation, and modals on the referencibility of their modified phrases. A particularly clear exposition of these and other surface effects on noun phrase acceptability for reference candidacy can be found in [Karttunen 76].


4. Detailed Example

Figure 4-1 provides some concrete examples of reference finding in context. The sentences are the first two of a Time article on discrimination suits brought against employers by employees. (For a further analysis of this text see the results section.) At this point we will gloss over many of the fine points necessary for a correct interpretation of the text (including details of the matching algorithm and other constraints on successful reference) and concentrate on the gross integration of phrases with context.

We indicate specific surface strings in quotes. Phrases pending integration are assigned a subscript corresponding to their line number in figure 4-1, i.e. the _network2. Successfully instantiated phrases are nodes with names of the form network:27.13

4.1 Index references

4.1.1 New instance

Initially context is empty. The first reference14 encountered is "NBC" (1a), which connects to a concept serving as the pivot for a description of an extensional entity. "NBC" as a lexical item points to a single conceptual item, NBC:0, and the task of computing a reference is trivial. In this case it amounts to the default case of instantiating a concept for which no referent was found in the discourse context. In general, proper nouns are no more unique labels of extensional entities than are ordinary nouns. In the appropriate context "NBC" could refer to the National Biscuit Collective, the Newark Board of Control, or New Bedford College, just as names like "John" have numerous referents which are reduced by context. Given multiple referents in the absence of context it is necessary either to suspend processing of the current phrase until more information is available or to rely upon some measure of strength of association or both.

Recognizing that "NBC" points to NBC:0 we create a manifestation, NBC:1, connected to NBC:0 by an equivalence relation. NBC:1 is a new terminal instance in a taxonomy. Its successive supertypes include NBC:0, broadcast-network:0, corporation:0, and business:0. Any of these nodes may have lexical connections. Links from the lexicon are access keys into conceptual space. Within limits to be discussed later (section 5.2.2.1) lexical items can be used to refer to instances of concepts that are subordinate or superordinate to the ones they index. Legitimate lexical references to NBC:1 in the established context include "NBC", "the network", "the corporation", and "the business".15 "Corporation" points to the concept corporation:0 which presumably has numerous sub-classes and instances, duPont:0, IBM:0, MacDonalds:0, and so on. The critical simplifying assumption that we make is that elements of context are preferred candidates for reference in the absence of overt signals to the contrary. Without context to act as a filter, the subordinates of such higher level concepts would present a bewildering array of possible references. According to our view, context functions to make available only the downward path that leads to NBC:0.

4.1.2 Existing instance

Figure 4-1 includes a number of references to an entity instance already in context. In 2a the lexical item "NBC" points to NBC:0 which has a manifestation, NBC:1, in context. NBC:6 is found to refer to NBC:1 and this reference is tied in appropriately to the representation of the phrase currently being interpreted, in this case "NBC has practiced sex-discrimination".

13 The numbers following the colon are arbitrary and are intended to distinguish different senses of lexical items and different instances of concepts. The reader is cautioned that the lexical portion of this arbitrary label is simply a convenience and any meaning acquired by the node is solely by virtue of its connection to other concepts.

14 As explained in 5.5 we treat every phrase as input to the reference algorithm. In case no explicit or implicit referent is found the algorithm creates a new instance.

15 That this last is possible in this particular knowledge base suggests that world knowledge is overgeneralized.
1. NBC has glumly maintained radio silence since the news broke: the network agreed to shell out $2 million to settle a sex discrimination suit brought by women employees in 1975.

2. As indirect acknowledgment that NBC has practiced sex discrimination some $1.6 million in raises and back pay will go to former and present employees.

\begin{verbatim}
1a NBC1 = NBC:0
1b the_network2 = NBC:0
1c agreed2 = (agree:13a on settled-suit:13a)
1d to_shell_out3 is pay:9 by NBC:0
to employees:15
= (result:13b on settled-suit:13a)
1e $2_million3 = money:10
1f settle3 = (settle:11 on settled-suit:13a)
by NBC:0
with employees:15
specializes suit:13 to settled-suit:13a
1g sex_discrimination4
= (illegal_act:13a
  on believe:13a by employees:15
  on settled-suit:13a)
by NBC:0 against employees:15
1h brought4 = (bring:14 on settled-suit:13a)
against NBC:0
1i employees5 = (employees:15 on NBC:0)
\end{verbatim}

\textbf{Figure 4-1:} Sample computed references
As another example, "discrimination?" references an instance in context of the act, discriminate:12 via the same computational route followed by NBC6. Discriminate:12 was originally instantiated as part of the analysis of the NP "sex discrimination suit". This resulted in an instance of a suit whose illegal act was discrimination. The "sex discrimination" portion of this phrase resulted in a concept paraphrasable as "The employees believe that NBC discriminates against employees on the basis of sex". Because it is located in a belief context one may choose to accept it or not. We have not implemented a belief management facility but one requirement of such a system would be to recognize that the later reference, "indirect acknowledgement that NBC has practiced sex discrimination", implies that it should update this belief on the part of the employees to the status of a legally accepted fact.

4.1.3 Implicit instance

References also occur to concepts which are not explicitly present in the discourse representation. Our solution to the problem of computing such references requires that an instance's entire taxonomic superset be added to context when the instance is created. This simplifies lexical disambiguation and, more importantly, provides an implicit expectation mechanism.

Business:0 is a structured concept that includes employer:0 and employee:1 as subparts. It is present in context because it is an element of the taxonomic superset of NBC:1. When we follow the lexical item "employee" to employee:1 we find that this concept, while accessible through context, has no instances created for it in this discourse representation. But by virtue of its accessibility, employee:1 has an implied existence and we try to instantiate this expected instance. The result can be seen in figure 4-2. Employee:1 resides on business:0. Business:0 does have an instance (transitively) in context, NBC:1. We create an instance of employee:1 on NBC:1 and create a ROLE relation (see section 3.4.1.3) between the instance and the generic concept. Creating an instance on a simple concept means that the concept has been elevated to the status of a structured concept. The creation of this instance of employee:1, fitting it into an expected slot on NBC:1, is the step equivalent to inferring that the employees are in fact employees of NBC. The only good reasons to make this assumption at this point in the text are the structured relation between business:0 and employee:1, the presence in context of a sub-instance of business:0, and the lack of any contradictory information.

This extremely important phenomenon reflects the reference algorithm's "belief" that discourse should be connected. It also sets clear limits on how much effort will be expended to make connections. We believe the reference process derives from the tendency of intelligent systems to integrate co-occurring phenomenon as closely as possible. This is a very primitive process whose success is largely a function of world knowledge. Much more active cognitive processes must be invoked when text appears disconnected.

In figure 4-1, 1b provides another example of an index term referencing an implicit concept. In addition it demonstrates one way in which context simplifies lexical disambiguation. "Bring" has a number of senses in Webster's [WNCD 74] including "to carry toward the place from which the action is observed", "PERSUADE", "ATTRACT", "COMPEL", and "PRODUCE". However the one that is contextually facilitated (figure 4-3) is INSTITUTE <～ legal action> in the suit:0 sub-context and refers to the initiation of the suit. Because this concept, initiate:24, was part of a concept in context this lexical sense is initially preferred to the others. The ultimate result is an instantiation of initiate:24, bring:14, on settled-suit:13a.

4.2 Reference by a more general term

4.2.1 Existing instance

In 1b (page 30) a general term, "the network", is used as a referent. The lexical label "network", connects to a superclass, network:0, which is followed down a contextually highlighted sequence of relations to NBC:1. Again, there are other senses of "network" possible, but only one that is in context. Similarly, there are other descendants of network:0 but only one path is accessible through context.

---

16 In our implementation this consists of simply adding a small number of nodes to a list.
**Figure 4-2:** Reference to an implicitly present concept
4.2.2 Subpart of existing instance

"Will go" in "$1.6 million will go to former and present employees" is particularly interesting. Will_go8 connects lexically (figure 4-4) through "go" to transfer:0 (in addition to numerous other concepts not in context) and down the taxonomy to pay:0 and thence to the instance, pay:9, which is the RESULT on settled-suit:13a. (In figure 4-4 the dotted lines indicate the links that result from the successful reference search.) A description of how pay:9 came to be instantiated is provided in the next section (4.3.1). The th role of pay:9 was employees:15. The th of will_go8 is employees:15 also, thus facilitating the match. However, the attempted match of will_go8 to pay:9 will result in a subpart of pay:9, pay:24, being created because pay:9 was for 2 million dollars and will_go8 for 1.6 million. This mechanism is covered in section 5.1. But the basic access to the candidate reference is another case of a lexical label for an ancestral concept used to refer to a descendant in context.

4.3 Reference by a specializing term

4.3.1 Implicit instance

The phrase "the network agreed to shell out $2 million" in the first sentence of 4-1 provides a complex example. Assume initially that settled-suit:13a in figure 4-5 is empty and pay:9 has not been instantiated. One of the elements of settled-suit:0 states that the settlement involved includes an agreement on the part of the person sued to compensate the initiator of the suit. "Shell out" lexically references pay:0. Pay:0 is not in context and as a result receives no preferential treatment with respect to other senses of "shell out". In this case there are no others. Having found a candidate sense of "shell out" we search for subordinate instances of pay:0 in context. There are none. Remember that pay:9 has not yet been instantiated. We then search for limited instances of its superclasses. Going up from pay:0 to compensate:0 we find that there is an implied instance of compensate:0 as the result:0 on settled-suit:0. This is the concept whose instance the phrase "shell out" references. We create the implied pay:9, an instance of pay:0, on settled-suit:13a and the result link is created because of the corresponding link between settled-suit:0 and result:0.

Compensate:0 as the concept between pay:0 and transfer:0 is an arguable representation. It is obviously possible to make the reference process more or less difficult by shifts in the details of the representation. "Pay", sense 4a [WNCD 74], is "to make compensation for" so this did not seem too illegitimate a decision. There are other possible structures. Pay:0 could be made directly subordinate to both compensate:0 and transfer:0. This possibility was ruled out because our implementation did not allow tangled hierarchies. "Pay" and "compensate" are also listed as synonyms with the shared meaning
Figure 4-4: Reference to a subquantity of an explicit concept
Figure 4-5: A more complex reference to an implicit concept
element of "to give money or an equivalent in return for something". It would be possible to create a node for this shared concept, making both \textbf{pay:0} and \textbf{compensate:0} subordinate to it. In any of these cases our reference algorithm would discover the referent of "shell out" to be the result, \textbf{pay:9}, on settled-suit:13a, so long as one of its senses was \textbf{pay:0} and \textbf{pay:0} had any of these plausible connections to \textbf{compensate:0}.

Despite any reservations about the particular structure of knowledge in this case, this sort of path to a referent seems critically important. The schema element referenced, \textbf{result:0}, is a conceptual instance of some concept which has many subclasses, any one of which may initiate a reference search leading through \textbf{result:0}, so long as \textbf{settled-suit:0} is in context. The organization of context to facilitate these paths is fundamental. We are following an input driven, bottom up processing strategy. Context provides a form of top-down filtering. This encourages usages that were not explicitly expected by the knowledge engineer.

4.3.2 Specialization of explicit concept by subpart

The referents of the arguments of a proposition are sought before the referent of the proposition itself. Thus, given the fragment "settle a sex discrimination suit" the first referent to be established is the one for "sex discrimination suit". Only then is one sought for "settle". One of the senses of "settle" is to come to an agreement in the context of \textbf{settled-suit:0}. Because this sense, \textbf{settle:0}, is an embedded concept, a referent of its enclosing concept is sought. This search leads to \textbf{suit:13} (already created because of the "suit" that is the object of "settle") which is then specialized to \textbf{settled-suit:13a} where \textbf{settle:11} is then instantiated.

This is a relatively simple process. When it is necessary to instantiate an instance of a concept that is a subpart of another, a referent for the enclosing concept is sought in order to have a proper place to put the subpart instantiation.

4.4 Computed result of example sentence

We follow a tactic of maximally connecting new concepts to preexisting ones. We only \textit{create} referenced or activated concepts, but we instantiate all of the links between these that are possible.

The analysis of the first sentence of the NBC text results in the structure of figure 4-6 and states roughly, "As a result of a suit brought against NBC by employees of NBC because the employees believed NBC practiced sex-discrimination against them, NBC agreed with the employees that NBC would pay $2 million to the employees as a settlement of the suit." Notice what has been "inferred" here. The suit was brought against NBC. The employees are employees of NBC. The employees believe NBC practiced discrimination against them. The agreement was made with the employees. The $2 million is to be paid to the employees.

The most important organizing concept in the analysis of the text is \textbf{settled-suit:13a}. \textbf{Discrimination:12} is the \textit{illegal-act} expected by \textbf{suit:13}. \textbf{Settle:11} specializes \textbf{suit:13} to a more specific organizing frame, \textbf{settled-suit:13a}, which has a slot for \textbf{settle:11}, \textbf{agree:8} and ultimately for \textbf{pay:9}. The interrelationships among these slots and their arguments provide the matrix in which the elements of the text can be assembled.

Inference is motivated by successful reference resolution. Once a reference has been found the matrix of relations that it inherits by virtue of its ROLE and ISA links are extended into the current representation insofar as there are existing concepts to be linked to. This motivation is important. Forward and backward inferencing from a concept is a potentially open-ended process. We only want to do those that are supported by the discourse and facilitated by context.
Figure 4-6: Discourse structure after one sentence
5. Reference Procedures

Our reference procedures can be conveniently split into three phases: reference candidate search, matching, and representation modification. Candidate search follows links from the lexicon through context to generate a set of potential referents. It is the contextual filtering during this phase that both reduces the number of candidates, restricts and selects implicit references, and provides the basic preference ordering over the results. Matching compares what is known about the input with what is known about the reference candidate to detect contradictions. This is essentially a comparison of their conceptual features and links to and from other concepts. Successful reference resolution may produce a variety of effects on the discourse representation ranging from no change at all to major reorganization. The most typical result is the addition of new information that augments concepts that are already elements of the discourse.

The top level procedure for integrating a proposition with the discourse seeks referents for the proposition as follows. The arguments of a proposition are resolved before the proposition as a whole. If resolution reaches an impasse because there are two or more equally plausible reference candidates, work on the current reference is suspended, to be resumed when more information is available.

We distinguish two classes of concepts as intermediate stages in the processing of lexical items. Lexical entries provide pointers to senses. Context may highlight particular senses, in which case they are preferred. The paths from a sense through context yield usages, the fragments of world knowledge that the sense may be tied to in this particular discourse structure. A usage is the immediate ancestor of a referent. If the referent is explicit then it has been previously instantiated and the link from the usage to the concept on the workspace can be followed. If the referent is implicit then no such link exists and a temporary candidate referent is created as a descendant of the usage.

The following sections first describe the input to the system and then present the three stages of reference search for simple, non-embedded concepts and the additional steps needed for the recursive procedure applicable to elements of structured concepts.

5.1 Input

Input to the system is in the form of lexical case predicates [Simmons 77]. Basically these are surface forms whose phrase boundaries and verb case arguments have been computed. Lexical ambiguity, except for that reduced by verb case argument naming, remains. Examples can be seen in figure 5-1.

NPs of the form N1-N2 are input as (N2 mod N1). Adj-N is treated similarly, (N mod Adj). Possessives, N1's-N2, and NPs of the form N2-of-N1 translate to (N2 of N1). Subordinate clauses, the-N-that-VP, become (N that VP), with the case argument position of N in the VP indicated using an asterisk. See the last example in figure 5-1, in which "suit" is the th of "bring".

5.2 Reference candidate search

The search for referents is outlined in figure 5-2. Usage and instance operations have been combined in step 2 of the figure. They will be described in separate sections.

5.2.1 Tracing lexical senses

We have ignored the morphological, syntactic and idiomatic aspects of our input. As a result the lexical entry for a word consists exclusively of pointers into world knowledge. Those concepts that are linked to a lexical entry are its senses. These correspond roughly to the lexicographer's decomposition in their granularity. Some of these senses are always available no matter what the context. Others are only accessible if the appropriate context has been established. It is always the case that senses in context are preferred to senses out of context.
Bill wanted it
⇒
(want ag Bill th it)

A rich man lived in a small town.
⇒
(live ag (man mod rich) loc (town mod small))

He was dissatisfied with his fortune
⇒
(dissatisfy ag he th (fortune of he))

The network paid $2 million to settle a discrimination suit brought by employees.
⇒
(pay ag network
 th (dollar quant 2000000)
 result (settle th (suit mod discriminate
 that (bring ag employees th *)))))

**Figure 5-1:** Sample translation to lexical predicates

1. Access lexical senses in context. If none, then access senses in **univ**.

2. Return the first referent candidate(s) found as follows, remembering the status of the search so that further candidates can be generated if necessary.
   a. Get explicit, subordinate instances.
   b. Get explicit, superordinate instances.
   c. Generate implicit, subordinate instances.
   d. Generate implicit, superordinate instances.

3. If no candidates were found and **univ** senses have not been tried, repeat 2 with these senses.

4. Finally, if no referent candidates have been found, create a new instance for a sense in context. If none were in context, create a new instance of one of the senses found in **univ**.

**Figure 5-2:** The reference process

If several senses are equally plausible we rely on the preference ordering provided by usage and referent search and ultimately on the matching process that compares, among other things, the semantic classes of the arguments of concepts. We have concentrated on the effects of changing context at the sentential level, but view intrasentential processing in a similar way. The standard approach to sense
disambiguation is to collect a set of senses and then reduce this set by comparing actual arguments with restrictions contained in the sense descriptions. An alternative view is to consider the arguments of a concept as augmenting context to facilitate the selection of the proper sense for the concept itself. As soon as we resolve a referent the appropriate modifications are made to context. These modifications naturally affect the processing of the remainder of the input phrase. Difficulties arise due to questions of control flow. In many cases two phrases work together to disambiguate each other rather than successful disambiguation of one then permitting the disambiguation of the second. Similar control problems occur in parsing and we would like to examine solutions similar to those of [Bobrow 80] with an eye toward integrating our reference procedures with a parsing algorithm.

It should be noted that initial lexical sense selection can be completely split off from reference search. We have combined them for three reasons. First, we believe that the same contextual filtering applies to sense selection as applies to reference search. Hayes [Hayes 77] has applied a form of contextual filtering to exactly this problem. Secondly, the lexicon provides our most basic entry into conceptual space. Given the absence of a parser, taking input from lexical predicates seems more plausible than beginning with preselected senses. And finally, sense selection cannot in general be completed without analysis at least to the level of reference interpretation.

5.2.2 Usage in context

Candidate senses generate a set of usages. A usage is simply a piece of conceptual knowledge accessed from a sense through context. It is the immediate ancestor of a referent. A sense which has no further connections into context is itself a usage. We distinguish sense and usage for two reasons. First we wish to separate the use of a word as a lexical item (a link to objects accessed through a dictionary) from its use as a pointer into a discourse. Senses are useful intermediaries, the result of directly traversing links from the lexicon. They provide entry points into the knowledge structures established in context. Usages are the extension of senses further into context. Words are more fluid in use than in isolation, yielding both more specific and more varied interpretations (usages) than senses considered alone would suggest.

The second reason to distinguish usages from senses springs from our desire to reflect the schematic and hierarchic nature of much of our knowledge. One of the things we have chosen for subordinate concepts to inherit is a potential lexical label. We are assuming numerous schematically linked instances of more general concepts. These subconcepts frequently do not have unique lexical labels. Our knowledge base is organized so that the lexicon points to general senses and then context suggests these subconcepts as more specific usages.

For example, there are many contexts in which an eating event occurs. We do not want all of them connected directly to the lexicon. This would be redundant as all are hierarchically related to the basic eat:0. Only this concept need be lexically indexed. At the same time these lower level schema parts are frequently referents of the lexical item "eat". While eat:0 is a sense of "eat", eat-breakfast:2 is a usage that is suggested by the breakfast schema. Again we require context to filter the numerous potential usages. There are any number of conceptual instances of eat:0. We have no interest in examining them unless there is some reason to do so. In fact we cannot examine most of them because the schemas of which they are parts are not elements of context, and the fact that they are more specific instances of eat:0 is unavailable. Given 5-2 alone

5-2. John ate an apple.

there is no reason to hypothesize any particular schema,17 but rather to create a simple instantiation of eat:0 ignoring for the moment its conceptual descendents. Only context suggests additional, more specific usages.

---

17 Assuming of course that we do not apply an eat-apple:0 schema or one devoted to John's unusual eating habits.
5.2.2.1 Superordinate usages

As explained in the introductory section on the specificity of reference (2.2.2), not all usages are senses or subordinates of senses. A possible usage is the result of checking the discourse representation for a concept being referred to by a lexical item. Implicit references are frequently more precisely specified in text than in their defining schemas. This is due to the fact that the schema represents a generalization across a set of examples. It is also possible, though less common, for a previously instantiated, general concept to be extended by a lexical item indexing a more specific concept. While normally reference search proceeds down hierarchically organized knowledge, in order to catch these cases it must look up.

This aspect of reference search has remained unclear. There are limits on how far up the knowledge tree one can look. In our implementation we simply marked certain concepts as ceilings on this sort of search. We call them basic concepts after KRL [Bobrow 77]. We have only used the ISA, EQUIV, and ROLE relations in setting up hierarchical data structures. The reasons for this are procedural. In all other aspects of reference search (in the downward direction) there exists a striking uniformity. It would seem we need to extend our representation to reflect more clearly the distinction between basic and other concepts along the lines suggested in [Rosch 77] and [Bobrow 77].

Upward search is limited by the following rules.

1. Upward search from a concept cannot descend from a node labeled disjoint, except to an instance or a conceptual instance (CI).

2. Upward search from a CI cannot descend through a partition which is disjoint with respect to the CI’s home partition. In figure 5-3 a candidate reference cannot be generated for A on D nor for C on E because D and E are disjoint subconcepts of F.

3. Upward search halts at basic concepts. Instances of the basic concept can be candidate references but search further up the taxonomy is blocked.

4. Once search has proceeded up a tree, descent is only permitted to an instantiated instance or a CI. In figure 5-4 from concept A we can access instance B and usage C, but we cannot follow link 3 through J to D.

The first and second restrictions are obvious. One should not ordinarily be able to use *reptile* to reference a mammal. Basic concepts were introduced to prevent clearly unprofitable searches. For example we don’t want to go up from apple:0 to thing:0 and then down through the entire thing:0 tree, which is almost certain to include elements in context. The last restriction is an empirically arrived at rule that reflects the Gricean conversational maxim to speak precisely [Grice 75]. It is present for the same reason that context restricts the accessible subordinates of a concept. While we want to allow a broad range of reference usage, limits are necessary if all elements of the preceding discourse are not to be constantly considered as referents.

5.2.2.2 Usage matching

During the search for usages some superficial tests based on surface information are applied as a quick filter. The most obvious of these are number checks (set vs. individual). Additionally it is important to know whether what is being discussed is an instance of some concept or the generic concept itself. This information is not necessarily available from the surface phrase. In the texts examined only one such instance appeared, 5-3.

5-3. Rufolo made the calculations that merchants make.

Rufolo was the agent of an instance of a concept whose agent is described by the generic concept "merchant". Recognizing the generic nature of this concept requires non-trivial interaction between the surface text, world knowledge, and context. We avoided this analysis by marking the input "merchant" predicate as generic:
Figure 5-3: Disjointness restriction

Figure 5-4: Descent restrictions
(do AG Rufolo
   TH (calculation THAT
      (do AG (merchant generic T) TH *))
)

The analysis results in a structure that states that Rufolo is the AG of a calculate that is tied by a ROLE link to the CI of calculate on the trading schema which was invoked by "merchant". As a result the conceptual representation of Rufolo is connected to merchant:0 via a ROLE link.

5.2.3 Embedded referents

The reference process is essentially applied recursively in the case of embedded concepts. We first search for a referent of the enclosing concept. Once that is done and context has been modified, the reference algorithm is continued with the embedded concept.

Figures 5-5 through 5-7 show some of the possible cases when searching for an embedded reference of lexical item "A". In case 1 (figure 5-5) SA1 has been instantiated on the workspace. The search for a referent of SA returns SA1 and we create an instantiation of A on SA1, A1, connected via a ROLE link to A.

```
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|ISA| context| lexicon
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>workspace</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 5-5: Case 1: SA is in context.
"A" instantiates A1 on SA1.

In case 2 (figure 5-6) only SB2 is in the workspace. After accessing sense A, a referent of its enclosing concept, SA, is sought. It has no descendents but it does have an ancestor in context, SB, which has an instance, SB2, in the workspace. SB2 is specialized from an SB to an SA by moving its ISA link. A ROLE instance of A is instantiated on SB2. Notice the enlargement of context to include SA.

In case 3 (figure 5-7) only C3 is in the workspace. No referent is found for SA, so a new instance, SA1, is created. The search for a referent of "A" yields C3 which gets a ROLE link to A in addition to its existing ISA link to C. By virtue of its new status as a slot filler of SA1, C3 may be integrated with other components of the workspace. The instantiation of SA1 leads to the inclusion of SA and SB in context.

5.2.4 Reference candidates

The search for usages and referents are in fact integrated. Both involve following ISA and ROLE links, usages being those concepts immediately above reference candidates hierarchically. Referents may already
be instantiated in the workspace. For those usages which have not previously been instantiated, we create a temporary instantiation as a potential implicit referent to match against the input.

5.3 Preference ordering of candidates

Figure 5-8 lists the different classes of sense, usage and referent and traces the possible connecting paths to a candidate referent. Each level is ordered left to right according to preference. Senses in context are preferred to those not in context. Usages in context are preferred to those not in context. Of these, subordinate usages are preferred to index usages which are preferred to superordinate usages. An index usage is a just a sense with no descendents in context. Existing referents are preferred to implicit referents which are preferred to entirely new instances.

When calculating preference, sense is more important than referent existence, which is more important than usage level. The initial filter on reference search provided by context during sense selection is the most significant. Once a sense is in hand, usages tied to existing instances (explicit referents) are sought, followed by a search for usages tied to implicit referents.

The legal possibilities can be sorted as in figure 5-9. Some of these possibilities that include senses in univ must be embedded usages because this is the only way that an explicit or implicit referent could be found when starting from a sense not in context. If the usages were not embedded then the senses, being hierarchically superior or equal to the usage, would necessarily be in context.

5.4 Reference testing

We use a straightforward matching algorithm to compare known aspects of the reference with the candidate referent. Because we are assuming that the preference ordering on candidate referents is reliable, the matching algorithm only checks for inconsistencies, as opposed to returning a probabilistic weight. The responses to *match input A against candidate B* are basically:
Figure 5-7: Case 3: C in context, C3 instantiated. "A" instantiates an SA and creates a ROLE link from C3 to A.
Figure 5-8: Relations between sense, usage, and referent

Sense in context
1. subordinate usage, explicit reference
2. index usage, explicit reference
3. subordinate usage, implicit reference
4. index usage, implicit reference

Sense in univ
5. embedded sub. usage, explicit reference
6. embedded index usage, explicit reference
7. superior usage, explicit reference
8. embedded sub. usage, implicit reference
9. embedded index usage, implicit reference
10. superior usage, implicit reference
11. index usage, new reference

Figure 5-9: Preference ordering of candidate referents

1. OK. No inconsistencies detected.
2. Failure. The two concepts are inconsistent.
3. Ok, conditionally. The two concepts are consistent if one views X (a part of A) as a Y (a part of B).
4. Ok, if A is a subpart of B, in the sense of an element of a set or a subquantity of a mass.

Matching is an open ended problem. Concepts have arguments that themselves have arguments, recursively. And, conversely, concepts are arguments of concepts that are arguments of concepts, recursively. Short of exploring most of memory this recursion must be halted at some point. We use a
numeric limit on the depth of recursion. We have arbitrarily chosen a matching depth of 2. When comparing two concepts we only extend the comparison to the point of assuring that the arguments of their arguments are of compatible types. Much more than this seemed of little use. In general a matching algorithm needs to be much more sophisticated in its analysis than ours for a deeper exploration to be worthwhile. We provide a very limited set of possible matching results. The more extensive the match, the more varied must be the set of responses that can arise from a suspected contradiction.

What constitutes sufficient difference for a mismatch is not obvious. Matching (person1 with (hair color red)) with (person2 with (hair color grey)) should yield something along the lines of "They don't match unless person1 can be considered to have aged." The ability to return this sort of conditional response seems important, but we have only implemented it for subpart matches.

Assume that we are matching concepts A and B. The question to be answered is "can A be a B?" First type is checked.\(^{18}\) Types match if one is an immediate descendant of one of the ancestors of the other or if it is a more remote descendant through CI links. Type mismatches occur if the concepts are members of separate taxonomic trees or if they are contained in taxonomic branches that have been marked disjoint (e.g., auto and airplane).

After the feature restrictions of A and B are compared to ensure that individuals do not match sets and extensional concepts do not match intensional ones, the matching algorithm is applied recursively to the filled slots of A and B. If there are no shared slots there may still be inconsistencies. B has a chain of ancestors from which it inherits restrictions on its arguments. A's arguments must satisfy these restriction templates.

The corresponding filled slots of A and B are matched. In many ways this comparison is best considered a reference search in which we check to see if the argument of B could be a referent of the argument of A. Frequently this has in fact been done previously and they are the same concept. We might have an input proposition of the form (A ag C) and resolve the reference of C to be D before attempting to resolve A. If our candidate match for A is (B ag D) then their arguments would trivially match.

Checking the restrictions placed on a concept by virtue of its role in other concepts is more difficult.

5-4. Jack bought a hat and Bill bought a hat. The hat that Jack owns is a size 7 and 7/8.

There are two reference candidates for the hat in the second sentence of 5-4. The hat that Jack bought should be preferred because the owning can be inferred from the first sentence, and the hat that Bill bought should be rejected because the agent of the owning that can be inferred does not match Jack. The procedure is as follows:

1. The input proposition is (hat THAT (own ag Jack th *)). The *** indicates that the hat is the th of the own.

2. The reference procedure tries to find references for the arguments of propositions before trying to resolve the proposition as a whole. In this case it will look for references for "Jack* and "hat", the arguments of the "own". "Jack" is resolved and "hat" results in two equally good referent candidates.

3. The own is then computed to be the implicit one on the buying by Jack in the workspace. The own by Bill is rejected because the ag slots fail to match.

4. Since the reference of "hat" was left hanging we return to it and there is now sufficient information to decide between the candidates.

One additional mechanism is necessitated by examples like sentence 5-5.

---

\(^{18}\) When the reference algorithm asks for A, the reference, to be matched against B, the referent candidate, it is unnecessary to compare their types, since this is what led to their consideration as matchee in the first place.
5-5. John had a bike. Bill wanted it. He gave it to him.

In our implementation the masculine pronominal references in the third sentence are handled by activation links functioning to enforce required condition checking. Activation links cause sufficient conceptual information to be instantiated for the matcher to decide between competing possibilities.

The want0 concept contains a negated have2 concept. I.e., if you want something, you don’t already have it. Given3 (from sentence 3) activates a similar implication that the agent of the giving does have the thing given. This have3 will find two reference candidates, the one from the first sentence and the one implied by the second sentence. Since the second is a negated concept, only the first will match, which results in the disambiguation of the subject *he* of the giving. The nonequality of *he* and *him* is then sufficient to disambiguate the *him* given that *he* is *John*. This inequality information is provided by surface syntax, since the second pronoun is not reflexive.

This would seem to be a reasonable place for the reference algorithm to interact with depth of inference. Rather than given0 activating have11, the inability of the reference process to decide between two equally good candidates might cause this more thorough exploration of preconditions.

Subpart recognition requires a procedural component for mass nouns that says “If both concepts are mass concepts that only disagree in terms of quantity, it is possible that the smaller is a subpart of the larger.” This effect propagates in an interesting way. If we are matching propositions of the form (A th X) and (A’ th Y) and Y is a subpart of X then the matcher suggests that A’ might be a subpart of A.

5.5 Representation modification

Concepts are integrated into the discourse structure in several ways.

1. A reference to an instance in the workspace is unified with this preexisting referent. This may mean the addition of arguments to the old concept or its inclusion as the argument of some new concept.

2. An implicit concept is instantiated on the organizing structure of which it is a part.

3. A reference may cause an existing referent to be specialized to a more precise description.

4. Entirely new concepts may be created or activated as the text unfolds, though typically these at least share arguments with preceding discourse elements.

5. Existing concepts may be subsumed, embedded as parts of a newly introduced organizing structure.

6. Implied relations may be instantiated between existing concepts and new ones according to their shared schemas.

The unification of a reference and an existing referent simply consists of the addition to the conceptual referent of any new information carried by the reference. No new concepts are created, only new relations between them.

Specialization requires moving the ISA link of a concept to a more specific concept. In addition the role links of the concept’s arguments are reoriented to the slots on the new, more specific type.

Implicit concepts are instantiated on the partition of which they are considered to be a part and their relations to previously instantiated portions of this defining frame are instantiated.

The subsumption of old concepts by new ones may occur in cases of activation. If, in the process of instantiating schema A, we activate concept B, then a referent for B is sought. When one is found it is made a part of the instantiation of A. The search for a referent of B proceeds in precisely the same way as the search for a referent of A, except there is no lexical disambiguation phase involved.
Whenever a new element of a schema is instantiated we attempt to relate it to other instantiated schema elements. Figure 5-10 outlines the process.

![Diagram of schema instantiation process]

**Figure 5-10**: Additional links added to an instantiated concept

Given input of the form \((P \text{ th } Y)\) followed by \((R \text{ a } X)\) the resulting structure would be the \(P1\) on the left of the figure. Then, by virtue of the relations on schema \(P\), \(P1\) would be linked to \(X1\) with an \(\text{ag}\) link and \(R1\) to \(Y1\) with a \(\text{b}\) link. Filling out relations in this way is important for later reference matching against these concepts. In conjunction with implicit reference it accounts for most of the inferences made by the system.

### 5.5.1 Activation during Instantiation

There are two case in which instantiating a concept may result in further reference search. The referred-to concept may be embedded, in which case an instance of its defining partition must be found or created. Additionally, the generic concept that is about to be instantiated may possess activation links. In both of these cases the reference algorithm is applied recursively to generate the additional instantiations.

We have also experimented with the effects of tense on activation. In the flight story (see section 6.3) there are certain questions that are easily answered by the reference algorithm if tense controls some activations. An event in the past tense activates its postconditions, making their subparts available for reference. This approach worked well in this simple example but a satisfactory general solution to the expansion and contraction of context in response to tense remains to be found.
5.6 Noun phrase integration

We use one device to assemble the components of noun phrases. Context is reduced to that provided by the hypothesized dominant member of the phrase, and referents for subordinate constituents are sought within this restricted context. If this search fails we sometimes reverse the roles of the constituents, i.e. a referent is sought for the dominant element within a context provided by the subordinate element. Samples from our texts can be seen in figure 5-11.

| small town       | (town mod small) |
| discrimination suit | (suit mod discriminate) |
| legal fees       | (fees mod legal) |
| his fortune      | (fortune of he) |
| rest of the money| (rest of money) |
| $2 million       | (money quant 2000000) |
| the calculations that merchants make | (calculation that (do ag (merchants generic t))) |
| suit brought by employees | (suit that (bring ag employees th *)) |

**Figure 5-11:** Noun phrase modification

1. \((A \text{ mod } B)\) : Seek a referent of \(B\) in the context of \(A\). If that fails, seek a referent of \(A\) in the context of \(B\).

2. \((A \text{ of } B)\) : Seek referent of \(A\) in the context of \(B\).

3. \((A \text{ that } (B \text{ r1 } *))\) : Seek referent of \(A\) as argument of \(B\). If there exist multiple possibilities, then \((B \text{ r1 } A)\) may reduce them to a single referent.

**Figure 5-12:** Noun phrase constituent referent search

The basic possibilities are listed in figure 5-12. In the case of input of the form \((A \text{ mod } B)\) we first look for a \(B\) in the context provided by \(A\). If that fails to produce a referent, depending on what kind of success we had in finding a referent for \(A\), we search for an \(A\) in the context provided by \(B\), unless \(B\) was a genuine adjective, in which case its representation will block reference search.

In the cases of the form \((A \text{ of } B)\), a referent is sought for \(A\) in the context of the referent of \(B\). This operation is very similar to Grosz' use of focus spaces in the the analysis of genitives [Grosz 79].

Dependent clauses are constructed on the partition of the dominant element of the phrase. Thus "The dog that bit the postman* and *the postman that was bitten by the dog* result in significantly different conceptual structures. In both cases, however, we look for an instance of \((\text{BITE ag DOG th POSTMAN})\) in the context. The respective input forms are \((\text{DOG that } (\text{BITE th POSTMAN ag } *))\) and \((\text{POSTMAN that } (\text{BITE ag DOG th } *))\).

This process does not provide a completely general solution to the problems of noun phrase computation but it is suggestive. In its reliance on existing knowledge structures, as opposed to special rules for noun pair construction, it is similar to the discussion of nominal compounds in [Brachman 78]. It accounts for the class of compounds Brachman calls concept-plus-dattr and many of the dattr-dattr cases. In a concept-plus-dattr compound, one element of the pair provides a slot in its definition into which the second element can fit. In the dattr-dattr compounds the two elements are both slot fillers in some third concept. For example the elements of the compound *lion house* fit into the theme and location roles of *live*, under one interpretation.
We are concerned with computing reference rather than with detailed representation, which is Brachman's primary interest. Thus, he presents only a cursory suggestion as to how one goes about computing compounds. The concept-dattr cases are relatively straightforward, one simply looks at all of the possible dattr slots on the concept and chooses the best one to account for the other element of the compound. In our case the dominant half of the pair determines a context and by the normal process of reference search its slot is filled by the subordinate half. The dattr-dattr case is more difficult as a bridging concept must be determined. Certain of these fit simply into our overall reference computation framework. In the "lion house" example, we would define "house" as "a building that animals live in" and the search for a referent of "lion" in the context of "house" would pick out the animal implicit on it.

The compound "hockey stick" is also easy if the hockey context includes as elements the equipment used to play it. If not, things become somewhat more tricky. Brachman suggests the use of "sports" information to conclude that the stick must be a piece of equipment used in the sport of hockey, because SPORT has among its conceptual slots EQUIPMENT which provides the only plausible link to STICK. This requires that "all possible datts of the verbal concept (including inherited ones) must be checked to isolate VALUE/RESTRICTIONS that cover the non-verb component"[p. 193]. This is not the way our algorithm works. If the representation included the fact that "sticks" can be "sports equipment" the desired result would be arrived at easily. To lean on Webster, again, this is one definition of stick [WNCD 74]. However, what if this sense of "stick" was not present? Some sort of analogous reasoning in a slightly expanded context to produce the sort of result that Brachman describes seems reasonable, but we do not have a mechanism to propose at this time.

There are more creative aspects to compounding than we have addressed. Take for example, "42 man squamish stick". Unless the reader is a devotee of ancient issues of Mad Magazine a plausible interpretation of this phrase may be hard to come by. A phrasal lexicon [Becker 75] might come in handy here, with an entry like "<number> man <game>" to suggest that squamish is a game and then the analysis of "stick" would proceed in the same manner as the impoverished "hockey" example.
6. Summary of Results

The reference algorithm we implemented has been tested on four fragments (figure 6-1), two generated for test purposes and two extracted from actual text. The results of our experiments are summarized in figure 6-2 and more complete breakdowns are presented in figures 6-4 through 6-10. If we exclude new instances, there were 48 referents to be found in our samples. Of these we correctly computed 41, or 85%. Of the explicit or implicit referents in the actual texts we correctly computed 80%.

John had a bike. Bill wanted it. He gave it to him.

Mary flew to Austin. The door opened. She exited and greeted her friend.

In a small town in Italy there once lived a certain rich man, Landolfo Rufolo. He was dissatisfied with his wealth and wanted to double it. Rufolo made the calculations that merchants make, purchased a large ship, loaded it with goods bought out of his own pocket, and sailed to Cyprus. [Boccaccio 72]

NBC has glumly maintained radio silence since the news broke last week: the network agreed to shell out $2 million to settle a sex discrimination suit brought by women employees in 1975. As indirect acknowledgement that NBC has practiced sex discrimination, some $1.6 million in raises and back pay will go to present and former employees. The rest of the money will go for legal fees and a staff to monitor compliance with the settlement. NBC also agreed to move women into 15% of its high-level jobs by 1981. NBC’s settlement is just the latest in a string of whopping payments in sex- and minority-discrimination suits. [Time 77]

**Figure 6-1:** Sample texts

<table>
<thead>
<tr>
<th>test text</th>
<th>real text</th>
</tr>
</thead>
<tbody>
<tr>
<td>textual</td>
<td>activ.</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
</tr>
<tr>
<td>New</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Existing</td>
<td></td>
</tr>
<tr>
<td>Explicit</td>
<td>7</td>
</tr>
<tr>
<td>Implicit</td>
<td>3</td>
</tr>
<tr>
<td>Implicit in text</td>
<td>3</td>
</tr>
<tr>
<td>Implicit in phrase</td>
<td>3</td>
</tr>
</tbody>
</table>

**Figure 6-2:** Summary of results by reference category. Failures are in parentheses.

Figure 6-2 distinguishes textual references, those sought on the basis of literal lexical elements of the input, from activated references which were activated by the instantiation of some concept into the context. Existing references which were activated by the instantiation of some concept into the context. Existing referents are further broken down into explicit and implicit referents. Referents implicit in a phrase are those that were discovered as part of the analysis of a noun phrase as described in section 5.6.
In the translation to predicates we have discarded some surface modifiers because their representation would have been contrived given our current understanding. The reduced texts can be seen in figure 6-3.

John had a bike. Bill wanted it. He gave it to him.

Mary flew to Austin. The door opened. She exited and greeted her friend.

In a small town in Italy there lived a rich man. He was dissatisfied with his wealth. He made the calculations that merchants make, purchased a large ship, loaded it with goods bought with his money, and sailed to Cyprus.

NBC has maintained silence since the news broke: the network agreed to pay $2 million to settle a discrimination suit brought by women employees. As acknowledgement that NBC has practiced discrimination, $1.6 million in raises will go to employees. The rest of the money will go for legal fees and a staff to monitor compliance with the settlement. NBC also agreed to move women into 15% of its high-level jobs by 1981.

**Figure 6-3:** Reduced texts

How one interprets "NBC has glumly maintained radio silence" is not obvious. What does it mean for a corporation to be glum? Have they really stopped radio broadcasting? These metaphorical extensions require sophisticated use of the subparts of structured concepts that are beyond the goals we have set.

While the reference algorithm that we describe seems to account for a great deal of the connectivity of text it is by no means a complete theory of text understanding. There are cooperating processes that must be available if lengthy spans of text are to be successfully interpreted. We will attend more to the reference failures than to the successes in this chapter, as the successes follow straightforwardly from the previously described algorithm. The failures indicate the directions in which the theory needs to be extended and suggest some of the needed interactions with other components of a text understanding system.

The major reason for the failure of references in this set of examples is that the intended referent is too deeply embedded in the partitioned structure of knowledge to be accessible through context. Typically the desired referent is on a partition which is on a partition which is in context. This is not sufficient to make it accessible. Obviously, by shifting knowledge structures around we could have made these references computable. We restricted ourselves to representations whose embedded structure represented the inherent dependencies we perceived, in order to avoid to some extent the temptation to construct a representation specifically designed for a particular text.

### 6.1 The trading voyage schema

The trading voyage example (figure 6-4) is interesting because we used an activation link to establish the connection between Rufolo wanting more money and his making the calculations that merchants make. **Want:** activates a planning concept, figure 6-5, which states that if someone wants something they will think up a schema which will get it and then execute the schema.\(^{19}\) It has conceptual slots labeled ACTOR, WANT, PLAN, KNOW, and ACT*. The ACTOR fills the agentive roles of all of the subacts. The WANT role is the concept that is lexically labeled "want" and which pulls PLAN-GRAPH into context. This knowledge structure captures some aspects of the use of plans in [Schank 77] and [Wilensky 78].

*Merchant* labels the agentive slot in the trading schema, so when Rufolo is connected via a ROLE link

---

\(^{19}\) The syntax of these conceptual descriptions is presented in Appendix A.
1 A rich man lived in a small town in Italy.
2 He was dissatisfied with his fortune.
3 He made the calculations that merchants make,
4 bought a large ship,
5 loaded it with goods purchased with his own money,
6 and sailed to Cyprus.

Reference Referent Computed?
--- --- ---
he2 man:1 yes
fortune2 wealth:1 on rich:1 yes
his2 man:1 yes
make3 plan:3 on merchant:0 yes
it5 ship:4 yes
his5 man:1 yes
money5 wealth:1 on rich:1 yes
load5 load:0 on transport:0 no
sailed6 move:0 on transport:0 no

**Figure 6-4:** A trading voyage initiation

```plaintext
(NODE plan-graph WITH (actor IS person):
    NODE LEX1 (want desire) want
        IS DISTINCT mental-state WITH (cause = "act*":
            initiate = "plan":
            ag = "actor":
            LEX1 (goal motive) th IS state):)
    NODE LEX1 (plan calculate) plan IS think WITH (th = "(know th)":
        result = "know":
        ag = "actor":)
    NODE know IS know WITH (LEX1 (plan calculation) th:
        initiate = "act*":
        ag = "actor":)
    NODE LEX1 (attempt) act* IS schema WITH (model = "(know th)"
        ag = "actor":
        exp-result = "(want th)")))
```

**Figure 6-5:** The planning schema

to merchant:0 he is also marked as the ag of the trading schema whose acts fit into the expected ACT* on PLAN-GRAPH, while *calculate* is the corresponding PLANning.

There are two failed references in this example. The load5 and sailed6 references should ideally be recognized as part of the transport subschema of the trading schema. The trade schema includes a transport as a subpart. However, the transport has not been instantiated when we encounter load5 and
sail6 and as a result they are not recognized as subparts of this CI of transport:0 on the trade schema. One possible tactic to take here would be to have load:0 activate transport, since it is generally the initial act in a transporting event. We have avoided this tactic because activation links seem ad hoc, given that we can give no clear statement of when a conceptual structure should include them that is independent of the text to be analyzed.

Sail is another problem. If the load5 did not activate transport:0 or was not present it would still be desirable to tie sail6 in as the going that effects the transportation of goods to another locality for purposes of trading them. This is a point of representation we do not understand. Transport:0 is in some sense a mover:0 and at the same time contains a subpart which is a move:0. Perhaps transport:0 is not taxonomically related to any higher level concept, but its kernel, move:11, functions to drag move:0 into context.

6.2 The want and give schemas

The activation mechanism was required to compute the references in sentence 3 of figure 6-6.

1 John had a bike.
2 Bill wanted it.
3 He gave it to him.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Referent</th>
<th>Computed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>it2</td>
<td>bike:1</td>
<td>yes</td>
</tr>
<tr>
<td>he3</td>
<td>John:1</td>
<td>yes</td>
</tr>
<tr>
<td>it3</td>
<td>bike:1</td>
<td>yes</td>
</tr>
<tr>
<td>him3</td>
<td>Bill:2</td>
<td>yes</td>
</tr>
</tbody>
</table>

Figure 6-6: A simple story with preconditions enforced via activation

*He* and *him* both have two possible referents, John:1 and Bill:2. The choice between these requires knowledge to the effect that one can only give away something that one has. Similarly, if you want something you probably don’t already have it. These conditions were attached as preconditions to their respective concepts and the have on give was linked to give with an activation link (figure 6-7). When want2 is encountered a negated have:0 is implied, which states that the person wanting the thing does not have it. It is not instantiated. Give3 is instantiated as give:3 and activates a have:0 whose ag is the same as the ag of the give:0. The reference search for this predicate finds the have:1 from the first sentence, forcing the ag of give:3 to be John. It cannot match the implied negated have of want:2 because of the modality conflict. In addition, the explicit have:1 is examined as a referent candidate first. Because the ps and the ag of the giving cannot be coreferential (as indicated by a NEQ link) the other pronoun can then be disambiguated.

This use of activation seems customized to this text. A more reasonable solution to the problem of referents that depend on interactions with a set of restrictions that are all implicit in the representation of the discourse, would be to increase the intelligence of the reference algorithm when confronted with two equally good candidates. A referent for the precondition of the giving would be sought when the arguments of the giving could not be resolved. There are two candidates implicit in the discourse representation but one is negated. This precondition search would determine John:1 to be the best candidate reference of he3.
(have WITH (ag IS person:
  th IS object))

(want WITH (ag IS person:
  th IS object:
  req IS have WITH (ag = ~ag:
    th = ^th:
    MODAL = NOT)))

(give WITH (ag IS person:
  th IS object:
  pa IS person WITH (NEG = ~ag):
  req IS have WITH (ag = ~ag:
    th = ^th)
  *A* = ~req))

Figure 6-7: The want and give schemas with activation

6.3 The flight schema

In the case of post-conditions on events in the example of figure 6-8 we have experimented with activation links sensitive to tense to instantiate the right information in context.

All of the referents of interest where computed except for the location of greet:23. This doesn't seem unreasonable. The reference algorithm cannot handle inferences of the form "If two events follow one another and no change of location is specified then the second event occurs at the location of the first." (Though it seems possible that a more sophisticated handling of context might.)

It is interesting to note the use of tense-based activations to tie the opening of the door and exiting with the arrival of the fly. The arrival of a fly-commercial:0, arrive:15, includes the scenario of the steward opening the door of the plane and the passengers exiting. The relation of open2 and exit3 to the arrival of the flight will only be determined if arrive:15 is in context. In order to accomplish this, the past tense of "flew" activates the final subevent of fly:0. That is, as events are closed off, we add their final subparts to context.

We do have reservations regarding this use of tense intermixed with the reference algorithm. We were experimenting with this use precisely because we are interested in the effects of tense on reference. Contrast the text of figure 6-8 to that of 6-9. There are several possible interpretations of this tense fragment. One is to assume that the flying was completed. The motivation to do so must arise from the recognition of the opening and exiting as subparts of the arrival, however. The other possibility is that the flying is still in progress and a) she left in midair or b) she got off of the plane at some intermediate stopover. In both cases something other than tense-based activation is required to knit the discourse representation together.

6.4 The legal suit schema

The Time magazine story (figure 6-10) is the most interesting of the set. We had originally intended to compute the references for the entire article. The primary difficulty is that as soon as things start to break down the context falls apart. If the context is not maintained in a reasonable state then the reference algorithm doesn't have a chance. The article points out a number of components needed in order to push further with reference resolution of the sort we have described.

We failed to compute the reference for discrimination3 because it was too deeply embedded. According to the world knowledge that the system possesses, the initiators of the suit believe that the target of the suit has committed some illegal act against them. In retrospect, placing the illegal-act of the suit on the
1 Mary flew to Austin.
2 The door opened.
3 She exited.
4 She greeted her friend.

5 Who opened the door, where?
6 What did she exit from?
7 Where is she?
8 Where did the greeting take place?

<table>
<thead>
<tr>
<th>Reference</th>
<th>Referent</th>
<th>Computed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(vehicle of flew1)</td>
<td>plane:1</td>
<td>yes</td>
</tr>
<tr>
<td>door2</td>
<td>door:2 of plane:1</td>
<td>yes</td>
</tr>
<tr>
<td>open2</td>
<td>open:2 of arrive:1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>of flew:1</td>
<td>yes</td>
</tr>
<tr>
<td>exit3</td>
<td>exit:3 of arrival:1</td>
<td>yes</td>
</tr>
<tr>
<td>she3</td>
<td>Mary:1</td>
<td>yes</td>
</tr>
<tr>
<td>she4</td>
<td>Mary:1</td>
<td>yes</td>
</tr>
<tr>
<td>her4</td>
<td>Mary:1</td>
<td>yes</td>
</tr>
<tr>
<td>friend4</td>
<td>friend:4 of Mary:1</td>
<td>yes</td>
</tr>
<tr>
<td>who5</td>
<td>stewardess:1 of flew:1</td>
<td>yes</td>
</tr>
<tr>
<td>where5</td>
<td>austin:1</td>
<td>yes</td>
</tr>
<tr>
<td>what6</td>
<td>plane:1</td>
<td>yes</td>
</tr>
<tr>
<td>where7</td>
<td>austin:1</td>
<td>yes</td>
</tr>
<tr>
<td>where8</td>
<td>austin:1</td>
<td>no</td>
</tr>
</tbody>
</table>

Figure 6-8: A simple story with activation

1 Mary was flying to Austin.
2 The door opened.
3 She exited.

Figure 6-9: Some effects of tense on reference
NBC has maintained silence since the news broke:
the network agreed to pay $2 million
to settle a discrimination suit
brought by women employees.
As acknowledgement that NBC has practiced discrimination,
$1.6 million in raises will go to employees.
The rest of the money will go for legal fees and
a staff to monitor compliance with the settlement.
NBC also agreed to move women into 15% of its high-level
jobs by 1981.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Referent</th>
<th>Computed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>news1</td>
<td>agree:2</td>
<td>yes</td>
</tr>
<tr>
<td>network2</td>
<td>NBC:1</td>
<td>yes</td>
</tr>
<tr>
<td>pay2</td>
<td>result:0 of settled-suit:3</td>
<td>yes</td>
</tr>
<tr>
<td>suit3</td>
<td>(specialize to settled-suit:3)</td>
<td>yes</td>
</tr>
<tr>
<td>settle3</td>
<td>settle:0 of settled-suit:3</td>
<td>yes</td>
</tr>
<tr>
<td>discrimination</td>
<td>illegal-act:0 (th) of believe:3 of settled-suit:3</td>
<td>no</td>
</tr>
<tr>
<td>employees4</td>
<td>employees:0 of NBC:1</td>
<td>yes</td>
</tr>
<tr>
<td>brought4</td>
<td>initiate:0 of settled-suit:3</td>
<td>yes</td>
</tr>
<tr>
<td>NBC5</td>
<td>NBC:1</td>
<td>yes</td>
</tr>
<tr>
<td>discrimina-</td>
<td>discrimination:3</td>
<td>yes</td>
</tr>
<tr>
<td>6-million6</td>
<td>2-million:2 (subpart)</td>
<td>yes</td>
</tr>
<tr>
<td>employees6</td>
<td>employees:4</td>
<td>yes</td>
</tr>
<tr>
<td>go6</td>
<td>pay:2 (subpart)</td>
<td>yes</td>
</tr>
<tr>
<td>rest7</td>
<td>2-million:2 (subpart)</td>
<td>no</td>
</tr>
<tr>
<td>fees7</td>
<td>fee:0 of lawyer:0</td>
<td>no</td>
</tr>
<tr>
<td>settlement8</td>
<td>settle:3</td>
<td>yes</td>
</tr>
<tr>
<td>NBC9</td>
<td>NBC:1</td>
<td>yes</td>
</tr>
<tr>
<td>agree9</td>
<td>agree:2 (parallel)</td>
<td>no</td>
</tr>
<tr>
<td>move9</td>
<td>contrast to not (promote:0 or hire:0) of discrimination:3</td>
<td>yes</td>
</tr>
<tr>
<td>jobs10</td>
<td>jobs:0 of NBC:1</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Figure 6-10:** Time magazine story beginning
belief concept was a mistake. In fact, this concept is central to the notion of a suit and needs to be readily available as a referent. This was an attempt to do too many things with the single device of partitioning. Partitioning in belief contexts was intended to be used to isolate believed concepts so that references to them and inferences about them would have to pass through the surrounding belief partition before accessing them, thus allowing their special treatment.

The failure to compute the correct referent for rest7 and agree9 is due to the lack of procedural components for concepts. Phrases of the form "the rest of the x" need to search for patterns of relations with restrictions that we currently cannot express. For example, in 6-2 "the rest" refers to a subset of a set of boys that has not yet been created. The existing conceptual anchor in this process is the other subset.

A reasonable representation of a sense of "rest" can be seen in figure 6-11.

```
  set/mass

  ISA

X

  ISA

  ISA

**************

* Y *

* *

**************

A1 T

-- UNION --

T A2

| EQ

disjoint

Z

<--- O---- "rest"
```

Figure 6-11: Conceptual sense for "rest"

Z is a subset of X, as is Y. Y must be an existing element of the discourse. The union of Y and Z is equal to X. The difficulty is the requirement that Y exist explicitly in the discourse. The X may be implicit or not even present, as it can be constructed. But Y must be present and our representation does not provide a means to express this sort of information.

As an experiment we added a simple form of procedural attachment to the matching algorithm, and provided the appropriate function for mass nouns. It worked well on the "money" references in this sample text. This procedural description was written in LISP however. What is needed is the appropriate level of "interpretive representation" to define the effects of the more functional lexical items on the reference procedure.

"Also" in line 9 is another instance of a functional concept. It introduces a parallel, but distinct,
concept. This is more strictly functional than "rest". ALSO(X) implies that there exists an instance in the discourses that X is like, except for the arguments specified.20 The possible approaches to handling such cases range from the addition of a SIMILAR link as part of the representation, to a more clearly functional kind of conceptual structure to be associated with such functional words.

"Legal fees" should map onto the expected fees that the female employees are to pay to their lawyers for bringing this case to court. In isolation this phrase might refer to the fees that NBC will pay. The significant clue here must be the parallel (which is in fact discovered by the reference algorithm but not used in this case) between the paying of money by NBC to the employees and the paying of the fees, with the employees indirectly benefiting. Recognizing this indirect benefit seems to be the difficult part of such a reference computation.

---

20Weber and Reiter [Webber 77] examine similar problems but do not include in their representation a means of specifying the requirement that a predicate already exist in the discourse structure.
7. Relation to Other Work

Text understanding is usually broken down into a set of hierarchically related cooperating processes, normally including lexical sense selection, syntactic parsing, semantic interpretation, anaphora and definite NP resolution, world knowledge application, and perhaps rhetorical or conversational analysis. It seems improbable that these various levels of language processing work according to radically different principles. We have proposed a small set of organizational principles and processing primitives that cut vertically across the problem of text understanding. As a result the proposals in this paper resonate to those of a diverse set of researchers. Useful surveys of much of the work on text understanding can be found in [Young 77] and, more recently, Hirst [Hirst 81] has surveyed research on reference.

7.1 Reference and context

7.1.1 Lockman and Klappholz

Lockman and Klappholz [Lockman 80] seem to share our basic outlook on text processing, that the fundamental operations over a text are attempts to integrate the current phrase with the context established by the preceding text. They are approaching the problem from a syntactic orientation and leave world-knowledge based inferencing as a black box. We approach textual integration as a fundamentally representation driven process and have described major portions of their black box, while ignoring surface syntactic effects. Additionally, we view reference as cutting through inference, in many cases preceding it.

Lockman and Klappholz define contextual reference resolution as

"... the phenomenon whereby an (any sort of) item in the semantic representation of a text has the property that for the text to be fully interpreted the reader must recover some (any sort of) intended, but not explicitly stated, connection (relation) between that item and some other item in the semantic representation of the text."

They qualify this by commenting that of course a reader is not in general going to make all and only those connections intended by the author. A connection is basically a chain of inferences tying two semantic interpretations together.

Their definition of reference is much like ours. We consider the process of referent search to begin at the lexical level in order that the set of potential lexical senses can be limited in the same way by context as are the reference candidates. We also consider reference to be the mapping from the reference, a lexical item or phrase, to some conceptual item which is its instantiation. This differs from their view of the reference mapping as a chain of inferences that relates interpreted semantic elements of the input to propositions literally present in the text. For us, reference is a mapping from an input phrase to its instantiation. For Lockman and Klappholz, reference is the set of inferences that connect the semantic interpretation of an input phrase to the existing discourse structure. The procedure that would be required to compute this chain is much more powerful than the reference algorithm that we propose. It allows for creative inferencing sufficient to account for the learning that takes place when one reads a text containing new material. Because we are attempting to describe the operation of the reference mechanism we have restricted our domain initially to mundane connections. For the references we do compute, our procedures produce inferential connections as a side effect of the reference process.

The structure of Lockman and Klappholz' context is based on the recognition of temporal continuation, conceptual subparts and the argument structure of the input. We construct a semantic representation of the discourse which generates similar relations between concepts. However the structure of their context is more a function of the presentation of the text than of dependencies prestored in conceptual representations. This aspect of the internal structure of context is largely missing from our context.
We disagree with them on some minor points. They describe the search for a *feasible connection* from sentence \( S_{n+1} \) of the text as giving special precedence to the immediately preceding \( S_n \), where the sentences \( S_1 \) through \( S_n \) of the text have been organized into a dependency structure. If a reference that is computed from \( S_n \) has a computed probability above some threshold, the search ceases. Otherwise a parallel search is undertaken through all of the \( S \)'s reachable in the context structure through \( S_n \) and the *best* candidate is selected. There are several problems with this description. It seems to be ignoring the necessity of sometimes computing the referents of the *parts* of \( S_{n+1} \) before it can be placed in relation to the rest of the text. While the algorithm they describe is designed to compute the dependencies between propositions represented by sentences of the text, it seems to contradict their earlier stance that all contextual reference be considered a unified phenomenon because *all of its manifestations in natural language texts are amenable to a unified method of resolution.*

We cannot handle all of what would be subsumed under their definition of contextual reference resolution. This is because we are separating the creation of well understood connections from creative problem solving. Consider the following.

7-2. I arrived late for work yesterday. The car wouldn't start.

7-3. I arrived late for work yesterday. The intersection of Brodie and McCartney Lane was flooded.

7-2 is their example 1-8. If our conceptual base were organized to facilitate the recognition of the fact that starting the car was the first element of a driving event (see 3.4.1.4) the resolution would be pretty straightforward. The driving would be found to refer to the travel that the arrival of the first sentence is the concluding event of. 7-3 is much more difficult. It must be inferred that the intersection was on the speaker's path to work and that the flooding caused a time consuming detour. This is outside the realm of mundane connectivity.

### 7.1.2 Grosz, Robinson, and Sidner

Our use of context in conjunction with reference resolution is similar in many ways to the work of Grosz [Grosz 77a, Grosz 77b], and Robinson [Robinson 80] on definite NP anaphora and VP reference in task oriented dialogues. Our context corresponds to a set of Grosz' focus spaces. Sidner [Sidner 79] has extended their work on discourse, by examining anaphora and definite NP reference in conversation.

Grosz describes an active focus space (Sidner introduces a similar immediate focus) which is the preferential location for pronominal and definite NP referents. Failure to find a referent in the active focus may result in a shift of focus to an element of the set of open focus spaces, which consists of previously active focus spaces. Providing this internal structure for context seems particularly to simplify pronominal reference computations. Perhaps even more importantly it provides a basis upon which to compute the dependencies of the concepts in the text based on their presentation as well as on the preexisting structure of knowledge, though this possibility has not been explored.

There are several major differences between our approach and theirs. The primary one is that we have made reference resolution the central component of the text integration process. This is a significant generalization of their work that results in considerable inferential side effects, particularly with respect to the expansions of implicit events. In addition there are technical differences. Robinson in [Robinson 80] describes an algorithm for verb phrase reference that searches forward from the expectations of the text for matching propositions in the input. This seems to work well in the task oriented dialogues with which she is dealing. In [Robinson 81] she distinguishes contextual constraints, which are drawn from the dialog and the subject area, from utterance constraints, which are derived from the syntax and semantics of the utterance. The primary path to verb referents is then top-down in the sense that contextual constraints are used "to find the place in the task that the utterance fits" and utterance constraints are used to limit alternatives. Our algorithm proceeds from the input text back into the discourse structure, relying only on the organization of knowledge, which combines the contextual constraints of the subject area with the semantics of the utterance.
For less directed discourse it appears better to search for connectivity from the input phrase back into context rather than from context out to potential input phrases. Robinson [Robinson 81] makes this same point when she states "In dialogs where less structure is provided by the task, a bottom-up search will clearly play a more central role. This search can be improved by more extensive reasoning based on the verb in the utterance." It is this bottom-up or data-driven search that we have been concerned with. Contextual constraints are a filter on the search for referents that match the hypothesized type of the input object.

7.1.3 Webber

Webber [Webber 78] in conjunction with Reiter [Webber 77] is concerned with the identification of possible antecedents of anaphoric expressions in the context of a representation based on formal logic. She separates the generation of a set of candidates from the task of choosing amongst them, which she assumes to require general world knowledge. Webber presents a set of rules for generating reference candidates for pronominal anaphora and verb phrase ellipsis.

Of particular interest in Webber's work is the effect of quantification on reference candidates and some rules for the generation of candidates given such expressions. This is an area we have ignored, relying instead upon world knowledge. Unlike Gross and Sidner, Weber does not attempt to account for implicit reference.

Webber and Reiter discuss the idea of "sloppy identity" in predicate reference [Webber 77]. The two possible interpretations of

7-4. Fred beats his wife. Ralph does too.

the second sentence of 7-4 ("Ralph beats his wife" or "Ralph beats Fred's wife") suggest that a device to support the lambda scoping described by Webber needs to be built into the representation and that solutions to the problems of "sloppy reference" are dependent in large part on the proper choice of representation. It would require significant revision to our representation system to support the "Ralph beats Ralph's wife" interpretation of this example.

7.1.4 Clark

Most of the implicatures that Clark [Clark 77] labels bridges are handled by our theory. Clark describes and categorizes the phenomenon of bridging. We present a method for the computation of bridges that ignores the given/new signals in the surface sentence. This seems to work because conceptual information is a stronger determinant of textual connectivity than are syntactic signals. See [Lockman 80] for an elaboration of this point.

Clarke describes four classes of bridge:

1. Direct reference
2. Indirect reference by association
3. Indirect reference by characterization
4. Reasons, causes, consequences, and concurrences

Direct reference bridges correspond to our explicit references, those that are already instantiated elements of the discourse representation. Most of these are relatively straightforward. Epithets are a convoluted subclass of direct reference and we do not understand why 7-5 sounds more acceptable than 7-6. Note that 7-7 seems more acceptable than 7-6, presumably because "greet" refers to a subpart of the meeting, facilitating the match of "man" to "rancher".

7-5. I met a man yesterday. The bastard stole all my money.
7-6. I met a man yesterday. The rancher stole all my money.

7-7. I met a man yesterday. The rancher greeted me warmly.

Most of Clark's indirect references correspond to the implicit references computed in our system. They are generally roles or subparts of schemas that have been established by the preceding text. They may be more complex combinations as in 7-8.

7-8. John died yesterday. The murderer got away.

Clark's description of the implicature involved in this example is "Some one person caused John to die; that one person is the antecedent of the murderer." A corresponding procedural account is necessary if we are to understand how such a connection might be computed. Figure 7-1 is a rough outline of the knowledge organization that supports the computation of the reference of "murderer* and the resulting representation.

In our view there are a number of different concepts subordinate to die:0. Some of these are parts of other schemas. The lexical item *murderer* picks out one of these other schemas, causing specialization of die:11 to the event caused by murderer:12. Assume that die:11 has already been instantiated by virtue of the conceptual interpretation of the first sentence. *Murderer* points to a role, murderer:1 in the murder:0 schema. When instantiating this event in order to provide the required schema on which to instantiate "the murderer", the activation arc (*A*) causes a referent of die:24 to be sought. By the normal process of reference search the previously instantiated die:11 is found to be the referent of die:24 and as a result is causally connected to murder:11. John:1 is made the th of murder:12 by virtue of the matrix of links on murder:0.

The bridges that Clark describes as "reasons, causes, consequences, and concurrences* fit less readily into the framework we describe. Those that are stereotypical are not a problem. It is those that correspond to the level of Schank and Abelson's plans [Schank 77] that are difficult. This problem is part of the larger one of how to characterize all of the variants of a schema. Suppose schema A has two possible outcomes. Two possible representations can be seen in figure 7-2. The choice between them seems to revolve around what implications the result has for the rest of the schema. If numerous inferences can be drawn relating to other elements of the schema then II seems more desirable because these relations can be stored on the respective subclasses of A. Additionally, this sort of representation ties in more naturally with the incremental construction of new schemas based on variant contexts.

7.1.5 The Yale group

Schank and his associates [Schank 77] have contributed a great deal to current theories of language understanding. We owe much to the ideas of scripts and plans, suggesting as they do that the proper place to look for referents, explicit and implicit, is in the larger structures invoked during text processing.

Recently [Schank 80] Schank has presented some general ideas on memory organization, particularly with reference to the integration of event memory in an accumulating body of knowledge. Our use of conceptual instances and ROLE arcs produces a very similar effect to that of memory organization packets (MOPs). MOPs provide for the sharing of packets of information among a number of schemas, just as conceptual instances do. Rather than a MOP-like strand, we use the device of partitioning. Perhaps more significantly, partitioning simplifies the access to contextually active memory, and provides a view of processing as a less active expectation process than that described by Schank.

7.2 Lexical sense selection

Hayes' dissertation on lexical sense disambiguation [Hayes 77] relies to a large extent on association-based techniques which are very similar to our use of context. In addition, while he is concerned only with sense selection, the techniques that he describes are also applicable to reference resolution. He does not
Figure 7-1: The integration of "murderer" with die:11
Figure 7-2: Two representations of schema results
create an explicit discourse context, concentrating instead on single sentences. The thrust of his collection of techniques is towards noun sense selection, with less attention given to verbs. The associations between verbs are not used to disambiguate them.

He proposes a set of classes of rules that contribute to sense selection. *Verb directed associations*, *selectional restrictions*, and *preference restrictions* are all a product of restrictions present in conceptual templates. *Verb directed associations* are "disambiguating rules based on relations between two of the participants in some event." For example given 7-9 there is a

7-9. The man was struck on the head.

rule associated with *strike* that states the the HIT:PART is a PARTOF the HIT:HITTEE. This relation can used by the disambiguating process to select the appropriate sense of *head* as a part of "the man". Hayes' selectional restrictions are the standard sort of slot filling restrictions and his preference restrictions help to order possible senses preferentially, without necessarily rejecting a filler that fails to satisfy them.

General association between the preceding context and possible senses is considered after the above constraints have been applied. This clearly differs from our order of processing, in which contextually restricted association is first used to establish candidate senses. Only after this step, as part of the matching process, do we check the internal relations between elements of concepts and the restrictions on the slot fillers of concepts. The computation of allowable associations in Hayes' system follows a restricted subset of the paths followed by our reference algorithm through hierarchically related concepts.

He introduces special processing for possessives and body parts. This processing is accounted for in part in our theory by the more general process of NP analysis within limited contexts. He points out an interesting restriction on references by body parts. The *body-part idiosyncracy rule* states that:

Whenever a noun phrase (that is not a pronoun) refers to a part of the body of an animal, which has been previously neither mentioned nor implied, then it must form the possessed part of a possessive phrase in which the possessor is the animal.

This is more a restriction on reference than sense selection. For example to take one of his examples, 7-10.

7-10. The man lifted the head.

it is relatively easy to construct a context in which the proper sense of *head* is the body-part sense (example 7-11),

7-11. The body lay dripping on the dock. The constable lifted the head.

but it does seem difficult to make the head be the head of the man doing the lifting. We have no mechanism in our system to prevent the resolution of *the head* as the implicit head of *the man* in example 7-10.

### 7.3 Coherence and text grammar

We have not been concerned with the question of how to decide if a text is coherent. The recognition that insofar as the reader is concerned it is not is assumed to be triggered by the exhaustion of resources.

Coherence has been defined in various ways. Simple repetition of discourse entities can be seen immediately to be inadequate and there have been various proposals for a more adequate definition. They are all characterized by a requirement of connectivity and some sort of structure according to "rhetorical predicates*. This is the label Grimes [Grimes 75] provides for a set of such relations. Meyer [Meyer 75]
has attempted to demonstrate that these relations can be used to explain recall of texts. Phillips [Phillips 75] explicitly required connectivity via paradigmatic relations, relations that were derived from the taxonomic organization of knowledge, and thematic relations that were derived from syntagmatic knowledge about the structure of events and event templates. Hobbs [Hobbs 79] has attempted to apply rules of coherence to simplify reference resolution.

Rhetorical predicates seem to be primarily descriptive. They are generalizations about the ways in which knowledge is organized and presented and not rules that can be applied as a part of the comprehension process. Some version of rhetorical predicates would seem useful in the generation of easily understood text. In other words, the best presentation strategy is not just to begin unrolling a data structure, even if its components are turned into correct English. Such a procedure may provide a certain amount of basic coherence because the resulting text focuses on some topic, assuming the data structure is a semantically connected one. Perhaps the most basic of event presentation strategies will apply, temporal succession. But questions of when to delete, when to elaborate, and when to violate temporal succession, all must be resolved at a higher level, by rules for generation.

Knowing a rhetorical rule does not of course mean that an individual is able to state it clearly, or even know that he possesses it. However most of the rules presented as rhetorical devices are not rules at all, but simply generalizations over textual phenomenon. For example, Hobbs presents three coherence relations, Elaboration, Parallel and Contrast and states that these can be used computationally to resolve reference problems. He presents an example of the application of the Elaboration relation to simplify the resolution of "he" in the example 7-12.

7-12. John can open Bill’s safe. He knows the combination.

His analysis seems overgeneralized in order to justify the use of the Elaboration relation. It is stretching things to state that "it is common knowledge that dialing the combination of some object causes it to open". Rather it is common knowledge that one of the ways of opening a safe, in fact one of the two most familiar methods, is dialing the combination. We would hypothesize that the first sentence of 7-12 brings in this knowledge as part the general knowledge of safes. This simplifies our selection of a sense for the lexical item "combination" and the knowing of the combination is recognized as a part of the opening. Since the opener of a safe is the one who must know the combination, "John" and "he" can be assumed to be coreferential. It is not by appeal to a general rhetorical rule that this match is made. It is the organization of knowledge to facilitate contextual reference that supports the computation.

We might restate Hobbs’ Elaboration relation by saying that when a conceptual object is introduced into context its parts are preferred reference candidates. The Parallel and Contrast relations have similar interpretations. In the Parallel relation, one instance of a concept is instantiated. This brings the defining concept into context and facilitates the processing of another such instance. The contrast relation works similarly, if less directly. In all of these cases, existing elements of context encourage the interpretation of later material in their terms. This is precisely what the dependence of our reference algorithm on context is designed to accomplish.

We do assume that there exist rules of rhetorical structure that a reader must bring to bear on some classes of text in order to understand it. How else to explain our ability to accept a logically flawed argument that fits a rhetorical template? Or at least recognize it as argument. The rules required for this sort of analysis would seem to be much more specific than those described by Hobbs.

7.4 Representation

Our representation is almost a caricature of common frame systems. We have attempted to reduce our system to those aspects required by the reference algorithm. We presented properties that we believe a formal level of representation must provide. We have used Hendrix’s [Hendrix 78] partitioned nets to implement our contextual restrictions and enablements, in a manner very similar to that of Grosz [Grosz 77a]. We have provided ROLE links between matching slot fillers in hierarchically related concepts, a notion that can be seen in several other representation systems, particularly Hayes’ binders [Hayes 77], which are bundles of ROLE arcs, and Brachman’s role and corefusal facility [Brachman 78]. Except for the functional effects of partitioning, our representation is a relatively simple frame system.
8. Future Work and Conclusions

We have tried to limit our research to a reasonably defined subpart of textual processing. This has been difficult because interpreting a text may call on such a wide range of processing power, including creative inference. In addition, the modules of a text processor are not independent. The parser must use world knowledge, perhaps even event memories, to complete its job. Likewise, syntactic information restricts and channels the application of world knowledge.

8.1 The structure of context

The primary extension to our system as it currently stands would be to add an internal structure to context to support the proper highlighting of its subelements. Currently the information necessary to compute failed referents is often present but too deeply embedded to be accessible. This lack of a context structure dependent on textual presentation is related to the problem of creating new dependencies between concepts based on the structure and focus of the surface text rather than stored knowledge. As a beginning we would incorporate some of the focus shifting rules described by Grosz [Grosz 77b] and Sidner [Sidner 79] and the effects of the goals of the actors in the text on such shifting operations, as described in [Carbonell 79]. In addition, if context had more structure and was sensitive to overt context switching clues in the surface text it might be possible to gain some insight into the transfer of elements of the workspace into knowledge space.

Another interesting feature of context is the activation of different classes of knowledge dependent on circumstances [Robinson 81, Charniak 79]. For example, some sorts of descriptive discourse might rely more strongly on the part/whole expansion of conceptual knowledge structures, as opposed to associated event knowledge. Such extensions depend on a more powerful view of activation, with activation links conditional on aspects of the discourse.

8.2 Intervening frame problem and schema selection

Because we have taken a very restricted view of context and its contents there are a number of referents which seem mundane in the sense that they are very closely related to the established context but are an element of a concept which has a conceptual instance which is a sub-partition of context. This relation is too distant in our system to be recognized as a plausible association. This is related to the problem of accessing referents that are deeply embedded in context.

There are a number of ways to approach this problem. The restriction that we would drop first is the requirement that a concept be a subconcept of only one other concept. We need to separate the structuring of conceptual objects from their contextual availability. If we call these contextual partitions subcontexts then a concept can be an element of a number of subcontexts. This will complicate the question of how subcontexts are added to context but seems a necessary feature of a contextually partitioned conceptual space.

We have not examined schema activation by other than lexical means. None of the clauses of 8-2 are sufficient alone to invoke a restaurant schema.

8-2. John sat down, ordered, ate his meal, paid the bill, and left.

Only in combination do they do so. The texts we have examined are not normally so spare, but similar problems arise. Rieger's "trigger trees" [Rieger 76, Rieger 78] suggest the kinds of extensions that need to be made to our activation methods for more extensive discourse processing to be successful. We have concentrated on reference search within a lexically established context rather than attempting a complete theory of how context is established.
8.3 Representation

Representation is still a fairly wide-open topic. We would like to adopt a more rigorously worked out representation scheme on which to impose contextual partitioning. The primary difficulty revolves around the fact that the two need to be integrated. A representation system that has contextual partitioning available will view the world differently than one that does not.

There exists in natural language a contrast between the apparent imprecision of metaphorical and vaguely metaphorical phrasing and the precision with which we trivially pick out the logical and temporal dependencies in text. The Time magazine article contains examples of this. "NBC has glumly maintained radio silence since the news broke." A reasonable interpretation of this sentence would seem to be that official representatives of NBC have refused to discuss the court case since the fact of the settlement was made public. This is a relatively deep inference to make on the basis of such a fuzzy piece of surface text. Is this what is constructed when the text is read? It requires knowledge along the lines of:

1. Corporations cannot speak, they use official spokespersons.
2. Maintaining silence means not communicating.
3. The news was significant enough that reporters probably queried NBC regarding their views on the court settlement.

That this level of sophisticated processing is necessary is not news. But even this difficult interpretation ignores "glumly" and "radio". "Glumly" does imply that NBC is not happy about their out of court settlement, but not in any direct way. We assume that "radio" has some effect on processing, since it is a broadcasting medium and NBC is a broadcasting network, perhaps facilitating access to the proper view of NBC. However, no representation that we are aware of addresses the question how the connotation of "glumly" or the conceptual structure of "radio" effects the interpretation of this phrase.

8.4 Parsing

In our extreme definition of reference we have covered much of the initial representation building involved in going from a syntactic structure to a semantic one. In order to integrate the reference algorithm into a parser in a satisfactory way a number of interactions need to be worked out. We envision a system in which the possible completion of a syntactic phrase triggers reference search. Extensions to phrases would always be possible requiring an approach like that described in [Chester 81]. Syntactic processes in addition to semantic processes would open and close context in order to generate the effects of tense, quantification, and various opaque contexts. It seems obvious that the order of appearance of clauses in surface text is important and should have an effect on reference processing. Thus it will not be reasonable to parse a sentence syntactically and then search for referents according to their appearance in a syntactic tree if this tree is the result of the application of transformational rules.

8.5 The representation of lexical senses

We have discussed the importance of more procedural representations for certain classes of words previously. This is a particularly important research area if we are to integrate reference resolution into the parsing process.

Lexical induction is another area that needs exploration if we are to attack significant size texts. Consider a slight variation of a previous example (2-3).

8-3. Ralph went to Guido's for lunch. As he was paying his bill, he tripped the waiter.

It may be the case that the reader does not know that Guido's is a restaurant in Galveston. He may infer this after reading the first sentence of 8-3. Or he may not, assuming that Guido is a friend of Ralph's and that Ralph went to his house for lunch. The second sentence will invalidate this interpretation and the restaurant interpretation will be generated. This recovery is beyond the bounds of our theory for two
reasons. In section 5.5 we discussed the results of a successfully computed reference. The recognition that a reference is not simply a new instance of some concept, but an invalidation of the current interpretation of the discourse requires powerful analysis procedures that we have not investigated. Additionally, both of these inferences, 1) that "Guido's" refers to the home of a friend of Ralph's named "Guido" and 2) that "Guido's" must be the name of a restaurant because Ralph ate lunch there, paid his bill and tripped the waiter, require knowledge about naming that muddies the clear but arbitrary boundary we have defined between conceptual knowledge and the lexicon.
8.6 Conclusion

When it comes to the meaning of anything, even the simplest word, then you must pause. Because there are two great categories of meaning, for ever separate. There is mob-meaning, and there is individual meaning. Take even the word bread. The mob-meaning is merely: stuff made with white flour into loaves that you eat. But take the individual meaning of the word bread: The white, the brown, the cornpone, the home-made, the smell of bread just out of the oven, the crust, the crumb, the unleavened bread, the shew-bread, the staff of life, sour-dough bread, cottage loaves, French bread, Viennese bread, black bread, a yesterday's loaf, rye, graham, barley, rolls, Bretzeln, Kringeln, scones, damper, matsen - there is no end to it all, and the word bread will take you to the ends of time and space, and far-off down avenues of memory. [Lawrence 66]

What we have demonstrated in this report is that if contextual sensitivity is integrated into a knowledge representation and a set of processing primitives, then the computation of referents is greatly simplified.

We have described three important aspects of reference resolution. A referent may be explicitly present in the discourse representation or it may be implicit as a part of the conceptual description of an ancestor of an element of context. A lexical item may be used with varying degrees of precision and its interpretation is fundamentally dependent on the existence of a taxonomic knowledge structure subject to contextual partitioning. Context may be used to help select the appropriate sense of a word, by giving preference to its senses in context.

The use of a context permits the extension of the notion of reference resolution to motivate a great deal of the textual integration required for intelligent processing of discourse. We find this property particularly appealing. There is a motivated set of processes in operation. Surface textual expressions trigger a series of operations that result in their instantiation. As a byproduct, additional organization may be imposed upon the representation of the discourse. It seems more productive to move from this mundane inferencing forward into the more creative applications of knowledge required for text comprehension than to initially postulate a powerful creative inference mechanism and then examine the ways in which it might be used to solve reference problems.
I. Database for the two real texts

This appendix includes the entire database used for the trading voyage and NBC suit texts.

The following is a rough outline of the input format for conceptual descriptions. Braces indicate optional constituents. The vertical bar (|) is an "or". The asterisk indicates one or more of the marked elements.

```
<def> = (<head> {<type>} {<parts>})
<head> = <addition> | <introduction>
<addition> = ADD <introduction>
<introduction> = <name>
               | <lex> ( <lexical-entry>* ) <global-name>
               | <lex> ( <lexical-entry>* )
               | NODE { <name> }
<lex> = LEX1 | LEX2
<type> = [ IS | ARE | = ] { THE | A }
         { DISTINCT } <superclass or node>
<parts> = WITH ( [ <subdef> ; ]* )
<subdef> = <def> | BLOCKER | NO-REF
<name> = <global-name> | '*' <local-name>
```

ARE indicates a set. *=* indicates that the slot name to the left is filled by the node to the right. THE labels an extensional concept. DISTINCT labels a concept that is a disjoint subclass of its superclass.

LEX1 indicates that the lexical link is always accessible. LEX2 covers those that are only accessible if the defining concept is in context.

Because the definitions are sensitive to ordering, some circular definitions require that ADD be used. One can define a concept by name without providing an expansion, define another concept which refers to the first and then ADD to the original concept some pieces referring to the second.

To refer to other concepts we use either global names or name prefixed with "*" to indicate the correct enclosing partition. One "*" indicates a pointer to another slot on this partition. Two indicate a pointer to a slot on the enclosing partition and so on. NODE marks a concept that is not connected by a link to its defining partition but merely resides there.

Every level of WITH, which begins the description of the subparts of the concept, indicates another, deeper level of partitioning.

BLOCKER labels concepts that block further upward search for referents. NO-REF is to prevent some adverbs and adjectives from invoking a reference search.
(black)
(caucasian)
(dummy-loc)
(female)
(male)
(mental-act)
(modal WITH (NO-REF))
(name ARE atom)
(not-thing)
(oriental)
(ADD (proceedings) enterprise)
(quantity WITH (NO-REF))
(relation WITH (th ; th2))
(part-of ARE DISTINCT relation WITH (part IS something ; whole IS something))
(rule)
(the-public ARE ALL. Of The people Of A country)
(time)
(tense)
(past ARE DISTINCT tense)
(future ARE DISTINCT tense)
(present ARE DISTINCT tense)
(unit ARE atom)
(almost ARE DISTINCT modal WITH (NO-REF))
(feature ARE relation WITH (value = th2))
(greater ARE relation WITH (ri IS number ; r2 IS number))
(legal-action ARE enterprise)
(loc ARE dummy-loc)
(loc-in ARE relation)
(not ARE DISTINCT modal WITH (NO-REF))
(true ARE DISTINCT modal WITH (NO-REF))
(number ARE quantity WITH (BLOCKER : NO-REF))
(size ARE quantity WITH (NO-REF))
(small ARE DISTINCT size WITH (NO-REF))
(sum ARE relation WITH (th1 : th2 : equal))
(at ARE relation WITH (th IS something : loc IS loc))
(in ARE DISTINCT relation WITH (th IS something : obj IS something))
(out ARE DISTINCT relation WITH (th IS something : obj IS something))
(authority ARE feature WITH (value IS quantity))
(country ARE DISTINCT loc WITH (name IS name))
(duration ARE time WITH (start-time IS time : end-time IS time))
(ADD loc-in WITH (LEX2 part IS loc : LEX2 whole IS loc))
(race ARE feature WITH (value IS either caucasian or black or oriental or indian))
(ADD relation WITH (inverse IS relation WITH (th1 = ~th2 : th2 = ~th1)))
(sex ARE feature WITH (value IS either male or...
female))

(something ARE loc WITH (  
    BLOCKER :  
    well-being :  
    life-line IS duration WITH (  
        LEX1 (creation) start-time :  
        LEX1 (destruction) end-time)))

(the-law ARE rule WITH (  
    authority IS authority))

(ADD loc WITH (  
    loc IS loc :  
    ADD IS loc-in WITH (  
        part = ** :  
        whole = ****loc)))

(state WITH (  
    BLOCKER :  
    th IS something :  
    duration IS duration :  
    loc IS loc))

(measure ARE DISTINCT state WITH (  
    th IS something :  
    value IS size))

(count ARE DISTINCT state WITH (  
    th ARE something :  
    value IS number))

(year ARE duration WITH (  
    name IS number))

(LEX1 (it) thing ARE DISTINCT something WITH (  
    BLOCKER))

(ADD state WITH (  
    enable = event))

(money ARE thing)

(medium ARE DISTINCT size WITH (  
    NO-REF))

(mass ARE DISTINCT something)

(air ARE DISTINCT mass)

(living-thing ARE DISTINCT something WITH (  
    BLOCKER :  
    life-line WITH (  
        LEX1 (birth) creation :  
        LEX1 (death) destruction)))

(ADD LEX1 (there) loc)

(large ARE DISTINCT size WITH (  
    NO-REF))

(event WITH (  
    BLOCKER :
result = state :
duration IS duration))

(Italy IS the DISTINCT country WITH (  
  name = (QUOTE italy))

(Cyprus IS the DISTINCT country)

(plant ARE DISTINCT living-thing)

(since WITH (  
  th2 IS event WITH (  
    end-time) :  
  th1 IS event WITH (  
    start-time = end-time of ~th2 :  
    end-time WITH (  
      before = NOW))))

(vehicle IS DISTINCT thing)

(water ARE DISTINCT mass)

(change ARE DISTINCT event WITH (  
  BLOCKER :  
  from IS state :  
  to IS state))

(former WITH (  
  obj IS thing :  
  event IS event WITH (  
    th = ~obj :  
    before = NOW)))

(month ARE duration WITH (  
  during IS year))

(percent IS quantity WITH (  
  th IS mass))

(ADD thing WITH (  
  x ARE * :  
  LEX2 (rest) ARE * WITH (  
    *A* = ~x) :  
  NODE IS sum WITH (  
    th1 = ~x :  
    th2 = ~rest :  
    equal = **)))

(total ARE relation WITH (  
  th IS mass :  
  quantity IS number))

(live ARE DISTINCT state WITH (  
  ag IS living-thing :  
  loc IS not-thing))

(animal ARE DISTINCT living-thing WITH (  
  LEX1 (home) IS loc :  
  LEX1 (eye) WITH (  
    partof = **) :  
  name IS name :  
  NODE LEX2 live IS DISTINCT state WITH (  
    th = **) :  
  unit IS measure))

(77)
loc = "home")

(current ARE time WITH (  
    obj IS thing :  
    event IS EITHER  
        (NODE IS event WITH (  
            ag = "obj")  
        or  
        (NODE IS event WITH (  
            th = "obj") WITH (  
                during = NOW)))))

(decrease ARE DISTINCT change WITH (  
    ag IS something :  
    from IS measure :  
    to IS measure :  
    NODE IS greater WITH (  
        r1 = " (from value) :  
        r2 = " (to value)")))

(increase ARE DISTINCT change WITH (  
    ag IS something :  
    from IS measure :  
    to IS measure :  
    NODE IS greater WITH (  
        r1 = " (to value) :  
        r2 = " (from value)")))

(LEXI (do) act ARE event WITH (  
    BLOCKER :  
    ag IS animal :  
    th IS something :  
    loc IS loc :  
    NODE IS at WITH (  
        th = "ag :  
        loc = "loc"))))

(day ARE duration WITH (  
    name IS number :  
    during IS month))

(ship IS DISTINCT vehicle)

(schema ARE act)

(person ARE DISTINCT animal)

(add person WITH (  
    LEXI (friend) IS DISTINCT person))

(october IS DISTINCT month)

(november IS DISTINCT month)

(march IS DISTINCT month)

(ADD LEXI (he) person)

(ADD LEXI (boat) ship)

(january IS DISTINCT month)

(february IS DISTINCT month)
(enterprise ARE schema)
(double ARE increase)
(dog ARE DISTINCT animal)
(date ARE day)
(cat ARE DISTINCT animal)
(car ARE DISTINCT vehicle)
(attack ARE act WITH (ag IS person :
LEX1 (target) pa))
(mental-state WITH (BLOCKER :
ag IS person :
th IS state))
(control ARE DISTINCT state WITH (ag IS person :
th ARE something))
(dissatisfy IS DISTINCT mental-state WITH (*
*IS = plan-graph :
ag = (of plan-graph actor) :
th IS state :
NODE IS change WITH (from = *th :
to = (of plan-graph want th))))
(eat ARE act WITH (ag IS animal :
LEX1 (food) th IS living-thing))
(halt ARE act WITH (ag IS person :
th IS act :
block = *th))
(maintain ARE act WITH (ag IS person :
start-time :
end-time :
th IS state WITH (start-time = *start-time :
end-time = *end-time)))
(LEX1 (go) move ARE DISTINCT act WITH (from IS loc :
to IS loc :
instr IS living-thing))
(LEX1 (leave climb-out) exit ARE DISTINCT move WITH (ag :
th IS thing :
result IS out WITH (th = *ag : obj = *th )))
(organize ARE act WITH (ag IS person :
LEX1 (staff) ARE person.
result IS act WITH (
  ag = ^^staff)))

(practice ARE act WITH (
  ag IS person :
  th IS act WITH (
    ag = ^ag)))

(think ARE mental-act WITH (
  ag IS person :
  result IS mental-state WITH (
    ag = ^ag)))

(agree ARE act WITH (
  ag IS person :
  at-time IS time :
  th IS act WITH (
    start-time WITH (
      after = ^at-time)))))

(believe ARE mental-state WITH (
  ag IS person :
  th IS EITHER
    (NODE IS state)
    or
    (NODE IS event)))

(choose ARE act WITH (
  ag :
  from :
  th ARE ^from :
  over ARE ^from WITH (  
    NEQ = ^th) :
  for IS act WITH (
    ag = ^ag :
    th = ^th) :
  NODE IS EITHER
    (NODE IS both
      (x of ^th)
      and
      (x of ^over WITH (  
        modal IS NOT)))
    or
    (NODE IS both
      (x of ^over)
      and
      (x of ^th WITH (  
        modal IS NOT)))
  WITH (cause = **))

(comply ARE act WITH (  
  ag IS person :
  act IS act WITH (  
    ag = ^ag) :
  WITH1 IS agree WITH ( 
    th = ^act)))

(damage ARE act WITH (  
  ag IS person :
  th IS something :
  result IS decrease WITH (  
    th IS well-being of ^th)))
(prepare ARE act WITH ( 
    ag IS person 
    th IS thing 
    for IS event WITH ( 
        th = ~th 
        precon = **) )
)

(refuse ARE act WITH ( 
    ag IS person 
    th IS act WITH ( 
        ag = ~ag 
        modal IS NOT 
        before = ~th )
    )
)

(transfer (go) ARE act WITH ( 
    ag IS person 
    th IS something 
    source : 
    goal : 
    at1 IS at WITH ( 
        th = ~th 
        loc = ~source 
        before = **) 
    at2 IS at WITH ( 
        th = ~th 
        loc = ~goal 
        before = ~at2 )
    )
)

(lose ARE DISTINCT change WITH ( 
    th IS something 
    pa IS person 
    from IS control WITH ( 
        ag = ~pa 
        th = ~th ) 
    to IS control WITH ( 
        ag = ~pa 
        th = ~th 
        modal IS NOT )))

(town ARE DISTINCT loc WITH ( 
    name IS name 
    loc IS country 
    LEX1 (inhabitant) ARE person WITH ( 
        home = **) 
    size IS size 
    NODE measure IS measure WITH ( 
        th = ** 
        value = ~size ) 
    NODE count IS count WITH ( 
        th = ~inhabitant 
        value) 
    LEX1 (population) value of count)

(pay (cost) ARE transfer WITH ( 
    ag IS person 
    for : 
    th IS money) )

(work ARE act WITH ( 
    ag : 
    for : 
    result IS pay WITH ( 
    )
)
ag = "for : 
recip = "ag : 
LEX1 (usage) th)))

(harm ARE damage WITH ( 
th IS person))

(ADD enterprise WITH ( 
result IS pay WITH ( 
LEX1 (payoff) th)))

(cook ARE prepare WITH ( 
LEX1 (cook) ag : 
for IS eat))

(ADD pay WITH ( 
LEX1 (compensation) th))

(ADD LEX1 (do) think)

(knuckle-under ARE agree)

(know ARE mental-state WITH ( 
LEX1 (thought) th))

(have ARE control)

(decide ARE choose)

(compensate ARE pay)

(own ARE have)

(information IS something WITH ( 
NODE IS know WITH ( 
th = **)))

(give ARE transfer WITH ( 
th IS thing : 
at2 IS own WITH ( 
ag = "th : 
th = "goal)))

(run ARE total WITH ( 
ag IS person : 
NODE IS pay : 
th = th of "pay))

(sense WITH ( 
ag IS animal : 
th IS something : 
result IS know WITH ( 
ag = "ag : 
th = "th)))

(rich WITH ( 
ag IS person WITH ( 
*Ag* = **) : 
LEX1 (wealth fortune) wealth ARE money : 
mag IS size : 
measure IS measure WITH ( 
th = "wealth : 
value = "mag))
NODE IS own WITH ( 
    th = ~wealth :
    ag = ~ag))

(use ARE DISTINCT act WITH ( 
    ag IS person :
    instr IS something :
    method IS act WITH ( 
        ag = ~ag :
        instr = ~instr :
        result = ~result) :
    NODE have IS have WITH ( 
        ag = ~ag :
        th = ~instr :
        enable = ~method)))

(business ARE person WITH ( 
    LEX1 (employer boss) = * :
    LEX1 (employee) ARE person :
    LEX1 (customer) ARE person :
    LEX1 (job) IS work WITH ( 
        ag = ~employee :
        for = ~employer) :
    purpose IS increase WITH ( 
        th IS money WITH ( 
            NODE IS own WITH ( 
                ag = ~boss :
                th = **) ))))

(NODE plan-graph WITH ( 
    actor IS person :
    NODE LEX1 (want desire) want 
        IS DISTINCT mental-state WITH ( 
            ag = ~actor :
            LEX1 (goal motive) th IS state) :
    NODE LEX1 (plan motive) plan IS think WITH ( 
        ag = ~actor) :
    NODE know IS know WITH ( 
        LEX1 (plan calculation) th :
        ag = ~actor) :
    NODE LEX1 (attempt act*) IS schema WITH ( 
        model = ~ (know th) :
        ag = ~actor :
        exp-result = ~ (want th)) :
    add want WITH ( 
        cause = ~act* :
        initiate = ~plan) :
    add plan WITH ( 
        th = ~ (know th) :
        result = ~know) :
    add know WITH ( 
        initiate = ~act*))

(exchange ARE DISTINCT act WITH ( 
    ag2 IS person :
    th2 IS something :
    ag IS person :
    condition ARE both 
        (NODE IS own WITH ( 
            ag = ~ag :
            th = ~th))
    and 
        (NODE IS own WITH ( 
            ag = ~ag2 :
            th = ~th2)))
ag = ~ag2 :
th = ~th2)

result ARE both

(NODE IS own WITH (    
ag = ~ag :
th = ~th2))

and

(NODE IS own WITH (    
ag = ~ag2 :
th = ~th)))

(LEX1 (go) go-by-vehicle ARE DISTINCT act WITH (    
instr IS vehicle :
ag IS person :
from IS loc :
to IS loc :
person IS own WITH (    
ag = ~ag :
th = ~instr) :
LEX1 depart IS DISTINCT act WITH (    
ag = ~ag :
from = ~from :
to = ~to :
instr = ~instr) :
move IS move WITH (    
ag = ~ag :
from = ~from :
to = ~to) :
LEX1 (arrive) IS DISTINCT act WITH (    
ag = ~ag :
to = ~to) :
add move WITH (    
then = ~arrive) :
add depart WITH (    
initiate = ~move))

(ADD go-by-vehicle WITH (    
depart WITH (    
NODE in1 IS in WITH (th = ~ag :
obj = ~instr) :
NODE LEX1 (board) IS move WITH (    
ag = ~ag : result = ~in1) :
NODE IS at WITH (th = ~ag : loc = ~from :
then = ~board) :
NODE IS at WITH (th = ~instr :
loc = ~from : then = ~board))
))

(transport ARE DISTINCT act WITH (    
ag IS person :
from IS loc :
to IS loc :
th IS loc :
instr IS vehicle :
exp WITH (    
NODE LEX1 (load) IS act WITH (    
ag = ~ag :
loc = ~from :
ths = ~th :
obj = ~instr) :
NODE move IS go-by-vehicle WITH (    
ag = ~ag :
from = ~from :
to = ^to:
    instr = ^instr:
    NODE LEX1 (unload) IS act WITH (ag = ^ag:
        loc = ^to:
        th = ^th:
        obj = ^instr)):

*A* = ~exp))

(sit ARE DISTINCT act WITH (ag IS person : loc IS thing))

(open ARE DISTINCT act WITH (ag IS person:
    from IS thing:
    th IS thing:
    NODE IS part-of WITH (part = ^th:
        whole = ^from)) )

(seat ARE DISTINCT thing)

(door ARE DISTINCT thing)

(plane ARE DISTINCT vehicle WITH (NODE door IS door:
    NODE seat IS seat:
    NODE IS part-of WITH (whole = ** : part = ^door):
    NODE IS part-of WITH (whole = ** : part = ^seat)))

(LEX1 (fly) fly-commercial ARE DISTINCT go-by-vehicle WITH (Lex1 (pilot) IS person:
    LEX1 (stewardess) IS person:
    FROM : TO :
    instr IS plane:
    *A* = ~instr:
    LEX1 (passenger) ag IS person:
    arrive WITH (th = ^instr : to = ^to:
        NODE IS at WITH (th = ^ag : loc = ^to):
        NODE IS at WITH (th = ^instr :
            loc = ^to):
        NODE exit IS exit WITH (loc = ^to:
            ag = ^ag : from = ^instr):
        NODE out IS out WITH (th = ^ag:
            obj = ^instr):
        ADD exit WITH (result = ^out):
        NODE IS open WITH (loc = ^to:
            ag = ^stewardess:
            th = ^ (instr door))):
    move WITH (NODE IS sit WITH (th = ^ag:
        loc = ^ (instr seat)):
    cause = ^arrive:
    medium IS air):
    future IS future WITH (#A* = ^depart):
    past IS past WITH (#A* = ^arrive))

(ADD business WITH (LEX1 (hire) IS change WITH (ag = ^employer:
    th IS person:
    from WITH (add ^th IS ^employee WITH (modal IS NOT):
        before = **):
to WITH (  
    add ^th IS ^employee) :
    after = "to" :
LEX1 (promote) IS change WITH (  
    ag = ^employer :
    th IS ^employee :
    from IS ^job WITH (  
        ag = ^th :
        th) :
    to IS ^job WITH (  
        NEQ = ^from :
        after = ^from :
        ag = ^th :
        th) :
    NODE IS greater WITH (  
        r1 = pay of th of ^to :
        r2 = pay of th of ^from))))

(oppose ARE act WITH (  
    ag :
    th IS act :
    NODE IS plan-graph WITH (  
        want WITH (  
            ag = ^ag :
            th IS halt WITH (  
                th = ^tgh))))))

(sail ARE go-by-vehicle WITH (  
    instr IS ship :
    medium IS water :
    LEX1 (dock) arrive))

(profession ARE business WITH (  
    LEX1 (professional) employer :
    LEX1 (fee) IS money :
    LEX1 (client) customer :
    pay IS pay WITH (  
        ag = ^customer :
        recip = ^ag :
        th = ^fee) :
    employee = ^employer))

(law ARE business WITH (  
    LEX1 (criminal) ARE person :
    employer = the-public :
    LEX1 (police) ARE ^employee :
    LEX1 (arrest) IS schema WITH (  
        ag = ^police :
        th = ^criminal)))

(legal WITH (  
    *A* = law))

(fight IS oppose WITH (  
    ag IS person :
    decide IS decide WITH (  
        th :  
            before = **) :
    th = th of ^decide))

(serve ARE give WITH (  
    cause IS eat WITH (  
        th) :  
            before = **)
th = th of "cause")

(sell ARE exchange WITH (th2 IS money))

(ADD own WITH (cause IS exchange WITH (ag2 = "ag :
  th2 = "th") :
LEX1 (possession) th :
start-time = (of * cause end-time)))

(communicate ARE transfer WITH (th IS information))

(buy ARE exchange WITH (th IS money))

(see ARE sense WITH (ag :
  instr = eye of "ag))

(say ARE communicate WITH (ag IS person :
pa IS person :
NODE know IS know WITH (th : ag = "ag :
  th = ~(know th) ))

(greet ARE DISTINCT say)

(observe ARE see)

(news ARE information)

(law ARE profession WITH (LEX1 (lawyer) employer))

(corporation ARE business)

(acknowledges IS say WITH (ag :
th WITH (modal IS TRUE)))

(Saks IS the corporation)

(Readers-Digest IS the corporation)

(Newsday IS the corporation)

(monitor ARE observe WITH (ag :
NODE IS act WITH (ag :
  th) :
  th = ~act (; or ag of ~act or th of ~ag))

(break ARE communicate WITH (th IS news :
NODE IS know WITH (th = ~th :
  ag = the-public)))
(television ARE thing WITH (  
    LEX2 (network) IS corporation :  
    LEX1 (broadcast) IS communicate WITH (  
        ag = "network :  
        instr = "**")
  )

(NBC IS the network of television)

(silence ARE state WITH (  
    ag IS person :  
    start-time :  
    end-time :  
    kernel IS communicate WITH (  
        ag = "ag :  
        start-time = "start-time :  
        end-time = "end-time :  
        modal IS NOT)))

(illegal-act ARE act WITH (  
    ag :  
    LEX2 (forbid) IS say WITH (  
        ag IS the-law :  
        th = "**" WITH (  
            modal IS NOT)) :  
    LEX1 (arrest) cause IS control WITH (  
        th = "ag :  
        ag ARE police of law)))

(restaurant ARE business WITH (  
    LEX1 (waiter waitress) ARE "employee :  
    LEX2 (cashier) IS "employee :  
    customer :  
    NODE IS eat WITH (  
        ag = "customer :  
        th) :
    NODE IS cook WITH (  
        ag IS "employee :  
        th = th of "eat) :
    NODE IS serve WITH (  
        ag = "waiter :  
        th = food of eat :  
        recip = "customer :  
        before = "eat) :
    NODE IS pay WITH (  
        ag = "customer :  
        for = food of "eat :  
        recip = "cashier) :
    add "eat WITH (  
        before = "pay)))

(command ARE say WITH (  
    ag IS person WITH (  
        NODE IS authority) :
    recip IS person WITH (  
        NODE IS authority) :
    NODE IS greater WITH (  
        th1 = authority of "ag :  
        th2 = authority of "recip) :
    th IS act WITH (  
        ag = "recip) :
    before = "th))

(sue (suit) ARE legal-action WITH (  

*A* = legal :
ag IS person :
LEX1 (defendant) th IS person :
LEX2 (bring) IS attack WITH ( 
  ag = \text{\textasciitilde}ag :
  th = \text{\textasciitilde}th :
  pa = \text{\textasciitilde}th :
  initiate = \text{\textasciitilde}) :
NODE IS believe WITH ( 
  ag = \text{\textasciitilde}ag :
  cause = \text{\textasciitilde}bring :
  th IS illegal-act WITH ( 
    ag = \text{\textasciitilde}th :
    cause IS harm WITH ( 
      th = \text{\textasciitilde}ag))))

(trade ARE DISTINCT plan-graph WITH ( 
  LEX1 (merchant) actor :
  want WITH ( 
    th IS measure WITH ( 
      th IS money :
      NODE increase IS increase WITH ( 
        from = \text{\textasciitilde}th :
        ag = (of trade actor)))) : 
  LEX1 (goods) goods ARE set-of something :
  act* WITH ( 
    NODE buy IS buy WITH ( 
      ag = \text{\textasciitilde}actor :
      th2 = \text{\textasciitilde}goods) :
    NODE sell IS sell WITH ( 
      ag = \text{\textasciitilde}actor :
      th = \text{\textasciitilde}goods) :
    NODE transport IS transport WITH ( 
      ag = \text{\textasciitilde}actor :
      from = \text{\textasciitilde} (buy loc) :
      to = \text{\textasciitilde} (sell loc) :
      then = \text{\textasciitilde}sell) :
    Add buy WITH ( 
      then = \text{\textasciitilde}transport) :
    result = \text{\textasciitilde} (sell result)) : 
  *A* = \text{\textasciitilde}act*))

(settled-suit ARE sue WITH ( 
  result IS compensate WITH ( 
    ag = \text{\textasciitilde}ag :
    recip = \text{\textasciitilde}WITH1) :
  LEX1 (settle) IS agree WITH ( 
    ag = \text{\textasciitilde}th :
    WITH1 = \text{\textasciitilde}ag :
    th = \text{\textasciitilde}result) :
  NODE IS believe WITH ( 
    cause = \text{\textasciitilde}settle :
    ag = \text{\textasciitilde}th :
    th IS lose WITH ( 
      ag = \text{\textasciitilde}ag :
      th = trial OF tried-suit))))

(discriminate ARE illegal-act WITH ( 
  ag IS business :
  for ARE employee OF \text{\textasciitilde}ag :
  against ARE person :
  choose IS choose WITH ( 
    ag = \text{\textasciitilde}ag :
against IS against:

for IS EITHER

(NODE IS hire ON ag WITH
   ag = ag:
   th = th)

or

(NODE IS promote ON ag WITH
   ag = ag:
   th = th)

or

(NODE IS increase ON ag WITH
   ag = ag:
   th = pay ON th))

NODE IS feature OF person WITH
   cause = choose))

(contested-suit IS sue WITH
   trial:
   court:
   th:
   result IS decide WITH
      ag = court:
      th IS act WITH
         ag = th):
      result IS command WITH
         ag = court:
         th = th)))

(lose-contested-suit IS contested-suit WITH
   ag:
   th:
   result WITH
      th IS transfer WITH
         recip = ag:
         ag = th:
         modal IS NOT:
         th))

(ADD sue WITH
   *A* = legal)
Bibliography

*Language, Memory, and Thought.* 

The Phrasal Lexicon. 

[Bellert 70] I. Bellert. 
On a condition of the coherence of text. 

[Bobrow 77] Daniel G. Bobrow and Terry Winograd. 
An overview of KRL, a knowledge representation language. 

KRL: another perspective. 

[Bobrow 80] Robert J. Bobrow and Bonnie L. Webber. 
Knowledge representation for syntactic/semantic processing. 

The Decameron. 
translated by G. H. McWilliam.

[Brachman 78] Ronald J. Brachman. 
*A Structural Paradigm for Representing Knowledge.* 

*Subjective understanding: computer models of belief systems.* 
Technical Report #150, Department of Computer Science, Yale University, January, 1979.

[Charniak 79] Eugene Charniak. 
With spoon in hand this must be the eating frame. 

[Chester 81] Daniel Chester. 
A parsing algorithm that extends phrases. 

[Clark 77] Herbert H. Clark. 
Bridging. 
[de Beaugrande 80]
Robert de Beaugrande.
*Text, Discourse, and Process.*

[Fahlman 78]
Scott E. Fahlman.
*NETL: a System for Representing and Using Real-world Knowledge.*

[Fahlman 80]
Scott E. Fahlman.
Design sketch for a million-element NETL machine.
In *Proceedings of the First Annual National Conference on Artificial Intelligence.*
American Association for Artificial Intelligence, August, 1980.

[Grice 75]
Paul Grice.
Logic and conversation.

[Grimes 75]
J. Grimes.
*The Thread of Discourse.*

[Grosz 77a]
Barbara J. Grosz.
*The representation and use of focus in dialogue understanding.*

[Grosz 77b]
Barbara J. Grosz.
The representation and use of focus in a system for understanding dialogue.
In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence.*
International Conferences on Artificial Intelligence, 1977.

[Grosz 79]
Barbara J. Grosz.
*Focusing and description in natural language dialogues.*
Technical note 185, SRI Artificial Intelligence Center, April, 1979.

[Hayes 77]
Phillip J. Hayes.
*Some association-based techniques for lexical disambiguation by machine.*

[Hendrix 75]
Gary G. Hendrix.
*Partitioned networks for the mathematical modeling of natural language semantics.*

[Hendrix 78]
Gary G. Hendrix.
Encoding knowledge in partitioned networks.

[Hirst 81]
Graeme Hirst.
Discourse-oriented anaphora resolution: a review.

[Hobbs 79]
Jerry R. Hobbs.
Coherence and Coreference.


Rieger 76]
Chuck Rieger.
Spontaneous computation in cognitive models.
Technical Report TR-459, Department of Computer Science, University of Maryland,
College Park, Maryland, 1976.

Rieger 78]
Chuck Rieger.
GRIND-1: First report on the Magic Grinder story comprehension project.

Robinson 80]
Ann Robinson.
The interpretation of verb phrases in dialog.

Robinson 81]
Ann E. Robinson.
Determining verb phrases referents in dialogs.

Rosch 77]
Eleanor Rosch.
Classification of real-world objects.
In P. N. Johnson-Laird and P. C. Wason (editors), Thinking. . Cambridge University

Schank 77]
Roger Schank and Robert Abelson.
Scripts Plans Goals and Understanding.

Schank 80]
Roger C. Schank.
Language and memory.

Shubert 78]
The structure and organization of a semantic net for comprehension and inference.
In Nicholas V. Findler (editor), Associative Networks - The Representation and Use of

Sidner 79]
Candace L. Sidner.
Focusing and discourse.

Simmons 77]
R. F. Simmons and D. Chester.
Inferences in quantified semantic networks.
In Proceedings 5th International Joint Conference on Artificial Intelligence, pages

Simmons 79a]
R. F. Simmons and D. Chester.
Relating sentences and semantic networks with clausal logic.
Technical Report, Department of Computer Science, University of Texas, Austin,
September, 1979.

Simmons 79b]
R. F. Simmons and A. Correia.
Rule forms for verse, sentences and story trees.
In Nicholas V. Findler (editor), Associative Networks - The Representation and Use of
[Stenning 78] Keith Stenning. 
Anaphora: an approach to pragmatics. 
In Morris Halle, Joan Bresnan and George A. Miller (editors), Linguistic Theory and 

Taking the Tube. 

[Webber 77] Bonnie Nash-Webber and Raymond Reiter. 
Anaphora and logical form: on formal meaning representations for natural language. 
Technical Report 36, Center of the Study of Reading, University of Illinois at Urbana- 
Champaign, Urbana, Illinois 61801, April, 1977.

[Webber 78] Bonnie Webber. 
A Formal Approach to Discourse Anaphora. 

Understanding Goal-based Stories. 

[WNCD 74] Henry Bosley Woolf (editor). 
Webster's New Collegiate Dictionary. 

What's in a link: foundations for semantic networks. 
In Daniel G. Bobrow and Allan Collins (editors), Representation and Understanding, 

[Young 77] Robert Young. 
Text understanding: a survey. 