The Influence of Time and Memory Constraints On The Resolution Of Structural Ambiguity

Susan Weber McRoy

Technical Report CSRI-209
September 1988

Computer Systems Research Institute
University of Toronto
Toronto, Canada
M5S 1A1

The Computer Systems Research Institute (CSRI) is an interdisciplinary group formed to conduct research and development relevant to computer systems and their application. It is an Institute within the Faculty of Applied Science & Engineering, and the Faculty of Arts & Sciences, at the University of Toronto, and is supported in part by the Natural Sciences and Engineering Research Council of Canada.

The Influence of Time and Memory Constraints on the Resolution of Structural Ambiguity

by

Susan Weber McRoy

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
November 1988

Revised from a thesis submitted in conformity with the requirements
for the degree of Master of Science, September 1987

Abstract

We propose a processing model that tries to give a reasonable account of how one might integrate several important psychological claims about the human sentence parsing mechanism (namely that processing is influenced by limitations on working memory and by a number of structural preferences, such as Right Association, Minimal Attachment, Revision as Last Resort, and verb-frame preferences). The starting point for this proposal is the Sausage Machine model (Frazier and Fodor, 1978; Fodor and Frazier, 1980), which gives a good account of memory constraints and sentence complexity, and incorporates most of the structural preferences we seek to include. From there, we attempt to overcome the more serious deficiencies of the Sausage Machine model, namely its dependence on ad hoc aspects of its grammar, and its omission of verb-frame preferences. The resulting model incorporates a principled theory of grammar, namely Government-Binding theory, includes mechanisms to handle lexical disambiguation and semantic processing in parallel with syntactic processing, and uses estimated timing information to resolve conflicting preferences.

Acknowledgements

Firstly, I thank my supervisor Graeme Hirst, for his support and guidance throughout this thesis work. He has been generous with his time even when he had very little time to be generous. I also thank Elan Dresher, my second reader, for his time and his helpful comments.

I am indebted to Barbara Brunson for long discussions of her interpretation of current Government-Binding theory into processing strategies, which are employed in the parser described in chapter four.

I also gratefully acknowledge Janet Dean Fodor and Lyn Frazier for their comments and suggestions about linguistic matters, Barb Brunson, Diane Horton, and Mark Catt for helping proofread this thesis, and to Jean-François Lamy for slaying LaTeX dragons. Many thanks to each of them.

I also acknowledge the financial support of the University of Toronto, the National Sciences and Engineering Research Council of Canada, and my parents.

Finally, I thank the people who have lifted my spirits even when my life looked its bleakest: my parents, Diane and Tom, Evan, and all my friends in the NLU group and DCS at large.

Contents

List of Figures

List of Abbreviations

AdvP Adverbial Phrase

ATN Augmented Transition Network

CP Complementizer phrase

Det Determiner

DP Determiner phrase

FBK Ford, Bresnan, and Kaplan (1982)

FF Frazier and Fodor (1978)

GB Government-Binding theory

HSPM Human sentence parsing mechanism

KB Knowledge base

LA Local Association

MA Minimal Attachment

MRS A knowledge representation system developed at Stanford (Russell, 1985)

N Noun

NP Noun phrase

Prep Preposition

PP Prepositional phrase

PSN Procedural Semantic Networks (Levesque and Mylopoulos, 1979)

PW A Polaroid Word process

RA Right Association

RALR Revision as Last Resort

S Sentence

SM Sausage Machine

V Verb

VP Verb phrase

Chapter 1

Introduction

1.1 General Motivation

Traditionally, only psychologists and linguists were interested in proposing models that tried to explain the human sentence parsing mechanism (HSPM). While computer scientists were worrying about how to build reasonably efficient programs that would assign to each sentence its "proper" structure (where "proper" was determined by the task at hand) these two groups were hoping to construct a theory of language that explained how and why people analyze strings of words the way they do; therefore their models had to not only assign the correct structures, they also had to be "principled" and "well-motivated".

Some more recent computer-science research in developing models for processing natural-language sentences has moved away from the original goal of just assigning proper structures in any way possible toward the goal of assigning these structures in a psychologically plausible way. This movement may have its roots in two fundamental problems: the difficulty introduced by ambiguity, and the difficulty associated with attempting to prove experimentally that a particular alleged feature of the HSPM is psychologically "real". The difficulty introduced by ambiguity has meant that sentence processing programs either had to produce all possible structures (which could be expensive, and wasteful because whoever generated the sentence presumably had only one meaning in mind), or they had to have some extra-grammatical decision mechanism for choosing the correct parse, hence the appeal to psychology. The difficulty in experimental verification caused by the preponderance of conflicting evidence (perhaps due to the variety of experimental methods available and the inherently subjective nature of many of these tests (Levelt, 1978)) has created a new motivation for building additional sentence processing systems. That is, implementing a proposed model can help advance psychological research by illustrating that the ideas behind the model are consistent and complete. Implementations may also help by pointing out problems or advantages of a particular model with regard to processing efficiency. Psychological reality requires efficiency, because a realistic model must respect any time and resource constraints known to limit the HSPM. Thus, building computational models that are based on psychological constraints can help us better evaluate psychological theories, and may help us discover superior processing methods.

1.2 The Proposed Model

This thesis presents a computational model of sentence processing that attempts to be as psychologically plausible as possible. The parsing model uses the general architecture of the Sausage Machine model (Frazier and Fodor, 1978), attempting to take advantage of several of its psychological merits without inheriting its dependence on ad hoc aspects of its grammar, a weakness that helped kill interest in the model before it could be fully tested. My parser extends the basic Sausage Machine architecture by providing an explicit account of the interaction between syntactic and semantic processing that emphasizes the importance of the relative time it takes to access or compute this information.

1.3 Important Issues

Although there are many psychological claims about the HSPM that are still in dispute, there are a few important claims for which any psychologically plausible sentence processor must account. In particular, one can assume that sentence processing is constrained by limitations on working memory, that the HSPM exhibits certain general preferences with respect to syntactic structures, and that syntactic and semantic processing are done in parallel. In what follows I will explain these assumptions in greater detail.

1.3.1 Memory Constraints

Among the better established claims about the HSPM is that it is constrained by limitations of its memory resources. It has been shown experimentally that working memory is limited in "size". Numerous researchers have found that working memory can only hold around seven units of information (Ebbinghaus, 1885; Miller, 1956; Hudson and Tanenhaus, 1985), while others have found that even fewer units of linguistic information may be held (Daneman and Carpenter, 1980; Zhang and Simon, 1985). The numbers vary because information is chunked into increasingly more complex units that take up less of working memory's capacity, although exactly how much less is not clear (Miller, 1956; Klatsky, 1980; Zhang and Simon, 1985). There is also evidence that processors and data must share the same memory capacity (Baddeley and Hitch, 1974; Daneman and Carpenter, 1980; Weber, Burt and Noll, 1986), so the relative complexity of the information and of the processes to be applied to it may have some effect on the number of units that can be stored. In addition, other researchers have argued that working memory appears to be limited by time (Baddeley, 1981). Time limitations might present an added complication to determining the "size" of working memory, except for the fact that it appears that the time constraint and the constraint on the number of units are in fact two sides of the same coin. Over time the contents of memory decay, but decay can be prevented indefinitely by rehearsal or reactivation. It is generally believed that size constraints arise from time constraints because working memory can rehearse only a limited number of elements without losing some to decay (Klatsky, 1980). Where controversy arises is in claims as to what the elements to be counted are; at times they could be subvocalized syllables, conceptual "chunks" of information, or some function of both (see Zhang and

Simon (1985) for a discussion). In any case, it is agreed that working memory is in some way limited, and, as long as one keeps one's vocabulary relatively simple, it is probably a relatively safe and natural simplifying assumption to count memory constraints in terms of the number of words and then of higher-order conceptual units. (A better, yet still crude, measure might incorporate some weight factor to account for processing complexity.) Thus, any reasonable model of the HSPM must be able to make all of its decisions without needing to have an unrealistically large amount of unstructured information in working memory at once.

Given that working memory is subject to rather severe limitations, it remains to be seen how these limitations affect real sentence processing. At the most basic level, memory limitations cause people to break sentences into more manageable segments for processing. Memory limitations also cause people to make educated guesses about the structure of the sentences they are processing, rather than trying to represent all possibilities at once.

Segmentation Effects

First, consider what happens when the HSPM has to decide how and when to break a text into chunks. Ideally, a reader or listener will break an incoming text into complete clausal units, but when under pressure from time or conceptual complexity this is not always possible. In these "higher cost" task environments, sequence length has been shown to cause people to settle for fragments of clauses such as phrases or partial clauses (Carroll, Tanenhaus and Bever, 1978). In addition, in situations where there is very little pressure, such as when clauses are relatively short and simple, a reader or listener might choose to chunk more than a single clause (Wingfield and Nolan, 1980). These two segmentation effects have been combined in a rather controversial general principle called "Local Association".

Local Association Local Association (LA) is a principle that states that terminal symbols (*i.e., words*) tend to be grouped either with the terminal symbols immediately to their left or with the terminal symbols immediately to their right, where "immediately" just means close enough so that all the intervening words fit in working memory at once. This type of local view differs from concepts called "locality" or "subjacency" which usually describe structural notions such as being dominated by the same S-node rather than a definition of localness based on input string length. LA was introduced in Frazier and Fodor (1978) and Fodor and Frazier (1980). They offer sentence (1) as an example of where LA takes precedence over other structural preferences.

(1) Though Martha claimed that she will be the first woman president yesterday she announced she'd rather be an astronaut.

In (1), "yesterday" is supposedly most naturally grouped with "she announced" rather than with "Martha claimed" because of LA, or with "she will be" because of semantic anomaly. This particular example is rather artificial, however, since presumably anyone who produced such a sentence would make their meaning clear by adding a pause or a comma before (or after) "yesterday". Other examples involve interpreting long lists of items such as in (2):

(2) John read the newspaper article, the card, the telegram, and the letter to Mary.

where "to Mary" prefers to be attached to "letter" instead of "read" as it would if it were closer to the main verb as in:

(3) John read the letter to Mary.

Despite Frazier and Fodor's apparent evidence, the existence of LA has been contested. Counterarguments have shown that much of the original data for LA can also be explained by the existence of a grammatical property of English that clausal complements are normally in the final position in their matrix clause (Ford, Bresnan and Kaplan, 1982). So, for example, in sentence (1), "yesterday" would preferably not have been attached to "claimed" because then the complement clause "that she will be the first woman president" would not have been the last constituent of the surrounding matrix clause "Though ... yesterday". Thus, it would appear that Ford et al. predict (4) will be very awkward, which agrees with the opinions of my informants as well as LA.

(4) Martha claimed that she will be leaving yesterday.

Ford et al. find, however, that their principle does not in general hold for adverbs; for example, their subjects showed no preference for either reading of sentence (5).

(5) Tom discussed Bill's dying yesterday.

To explain the lack of expected preference, they suggest that adverbs may be attached to the top S node rather than to a VP or they "may be treated differently from other categories because they can occur in so many places within a structure" (Ford et al. 1982: 784). This suggestion greatly weakens their explanation for the difficulty of (4).

Although the argument for a special grammatical property may have some merit, it does not in itself invalidate the existence of LA because it neither explains why this property exists in English, nor does it account for the other examples. It could not, for example, explain the apparent LA effect in (6), where I claim that it is more natural to associate "in New York City" with "he just met" than with "John went dancing".[1]

(6) John went dancing with a young deaf girl that he just met in New York City.

LA may also be the reason why in (7), once one rejects the attachment of "yesterday" to "plan to" for semantic reasons, the next most likely choice seems to be to attach "yesterday" to "he told me" (there may be some semantic effects at work here as well).

(7) Though John supposedly wrote the article on terrorists that plan to bomb hospitals yesterday he told me that he never writes about unhappy subjects.

---

[1] The clausal complement restriction does not apply here because relative clauses on noun phrases need not be clause-final, as shown by sentences like:

(i) John will buy the new book that Chomsky has written tomorrow.

Admittedly both these examples suffer from a certain amount of unnaturalness because presumably speakers and writers would use punctuation to make their meaning clear, but if one had to guess where to add punctuation, LA clearly would have some influence on preferred phrasing. The most serious criticism of LA is that it is not an absolute principle. As we will see later, there are many other influences, such as semantic or pragmatic effects, that influence preferred structure as well.

Ambiguity Storage Effects

As mentioned earlier, memory limitations also affect how the HSPM handles temporary ambiguity. Fragments like "that fish" are temporarily ambiguous because, without more information, it is unclear as to whether the fragment is a simple noun phrase, or the beginning of a clausal complement such as "that fish like" or a relative clause as in "people that fish". Since capacity is limited, the HSPM cannot afford to keep a lot of analyses going at once. One way to keep analyses open is to hold off deciding about what to do with a piece of information and not incorporate it into a bigger unit until it is clear which one of the available options is the correct one (Marcus, 1980). However, because of memory limitations, people can only keep a limited amount of material unstructured. Eventually, a decision is going to have to be made, and sometimes it will have to be made before the crucial piece of information has been received. Marcus (1980) claimed that this was a source of "garden-path errors"; however, other accounts of garden-path errors have focused on the violation of lexical (Milne, 1982) or structural (Frazier and Fodor, 1978) preferences. These preferences, however, may be called into play exactly because the HSPM has learned its limitations. In fact, others (Frazier, 1978; Frazier and Fodor, 1978; Marslen-Wilson and Tyler, 1980; Hirst, 1984; Frazier and Rayner, 1982; Rayner, Carlson and Frazier, 1983; Abney and Cole, 1985) point out that people do not delay decisions until they use up all their processing capacity. Thus, finding a plausible reading of (8) is difficult, even though information in the second prepositional phrase indicates the correct attachment of the first one (Abney and Cole, 1985).

(8) Hang the sign on the elephant on the flagpole.

When faced with an ambiguity, people apparently take some measure of the preferences at hand, and if there is sufficient evidence to make a choice, they make it and continue. Later, if necessary, they can retract this choice and try an alternative, although this may be difficult. Thus, memory limitations, and perhaps some built-in respect for these limitations, influences the HSPM's decisions about sentence structure.

1.3.2 Structural Preferences

In addition to structural preferences that arise as a direct result of memory limitations, there appear to be some whose explanation is not so obvious. For the most part, one could argue that these parsing preferences arise because they make the job of parsing sentences easier or more efficient. Such arguments often involve concepts such as "Right Association", "Minimal Attachment", "Revision as Last Resort", and "verb-frame preferences". I will discuss each of these principles in turn.

Right Association

The principle of Right Association (RA) states that optimally, terminal symbols will be attached to the lowest nonterminal node that is on the right-most branch of the current structure; that is, they will be grouped with the terminal symbols immediately to their left (Kimball, 1973; Frazier and Fodor, 1978; Wanner, 1980; Fodor and Frazier, 1980). For example, in a sentence like **(9)**

(9) Joe called the friend who smashed his new car up.

the particle "up" is usually associated with the verb "smashed" rather than with "called", although both attachments are possible (Fodor and Frazier, 1980). This property is very similar to LA, which was described earlier, partly because LA was first proposed as a generalization of RA:

> We started with Kimball's Right Association, argued that there were data it could not account for, revised its content, and renamed it Local Association. (Fodor and Frazier 1980: 425)

RA differs from LA, however, in that LA is only meant to describe the phenomena resulting from memory limitations, and hence does not specify any preference among the set of locally accessible attachments. Since there appear to be preferences to right-associate even within very short sentences, such as **(10)**

(10) Joe said Bill died yesterday.

where "yesterday" optimally modifies "died" instead of "said" (Wanner, 1980), RA has been revived as a separate principle. If there is to be any doubt as to whether these principles are in fact distinct, it is the existence of LA that should be questioned, because the examples of LA that are not also examples of RA are generally unnatural. RA in itself is an economizing principle, because a parser following RA will always consider attaching a new word to the nonterminal node nearest the one it is already working on, *i.e.*, either directly to the current nonterminal, or to the node nearest to it on the right; hence RA will help minimize both the number of attention shifts between nodes and the number of nodes that must be kept easily accessible (Wanner, 1980; Fodor and Frazier, 1980).

Minimal Attachment

Minimal Attachment (MA), as described by Kimball (1973), Frazier (1978), Frazier and Fodor (1978) and Fodor and Frazier (1980), requires that optimally a terminal symbol is to be attached into a parse tree with the fewest possible number of new non-terminal nodes linking it with the nodes already in the tree. The original description of MA also assumes that the grammar is such that the favoured attachments are represented using the fewest number of rules and the fewest number of levels of intermediate nodes, such as in the following context-free grammar:

S → NP VP
VP → V NP (PP)
NP → Det N
NP → NP PP

Note that this grammar includes an asymmetry: a PP modifying a verb is attached as a sister to the verb within its VP, whereas a PP modifying a noun is attached as a sister to an NP that contains the noun. (This asymmetry is not required by the grammatical theory.) Given this grammar, however, a parser looking for the preferred attachment need only count the number of intermediate nodes it would have to add in order to attach an incoming phrase for each alternative, and then choose the attachment that requires fewest new nodes. Consider sentence **(11)** for example:

(11) Joe bought the book for Susan.

Once the parser has built a syntactic structure for "Joe bought the book" such as in figure 1.1 and it receives the PP "for Susan", MA says the PP should be attached directly to the VP node as in figure 1.2 (which is the preferred reading). To have attached it to the NP would have required accessing the rule NP → NP PP and creating an extra NP node (see figure 1.3). Thus we see that a minimally attached phrase can be computed most quickly and will make the best use of the limited capacity of working memory.

Note that in **(11)**, MA acts in violation of RA. From this and similar sentences, one might predict that MA always overrides RA; however recall that in sentence **(2)**, which is superficially similar to **(11)**, the PP "to Mary" associates with the nearby NP "the letter" rather than minimally attaching to the VP headed by "read" as predicted by LA. Furthermore, in sentence **(7)** both MA and RA have been violated. It has thus been argued that LA constrains both MA and RA and hence explains the relationship between MA and RA. In particular, RA may override MA when MA is not LA (Fodor and Frazier, 1980).

Support for MA comes from experiments described in Frazier (1978), that show how MA operates in many other constructions. MA can also account for such well-noted phenomena as the preference for analyzing words or phrases as determiners, main verbs, direct objects, or conjuncts, rather than as the start of a complement or relative clause, making the first sentence in each pair of sentences below easier to understand than the second:

(a) That sick sheep should be kept in the barn.

(b) That sick sheep should be kept in the barn is obvious.

(a) The horse raced past the barn and fell.

(b) The horse raced past the barn fell.

(a) The girl knew the boy from school.

(b) The girl knew the boy liked school.

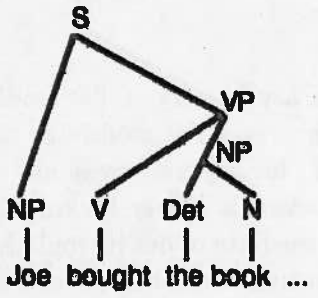

Figure 1.1: Initial Structure

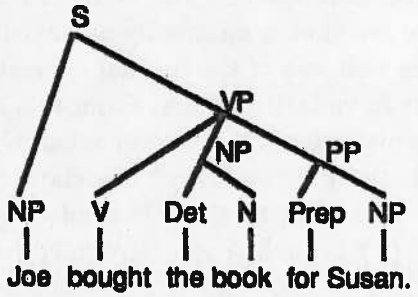

Figure 1.2: A Minimal Attachment to the Structure of Figure 1.1

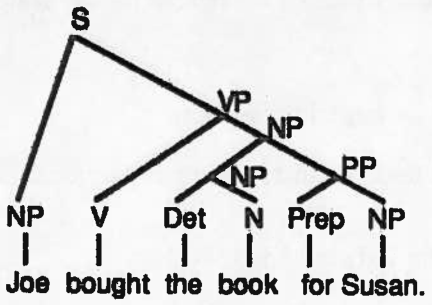

Figure 1.3: A Non-minimal Attachment to the Structure of Figure 1.1

(a) The man the girl and the boy met told jokes and laughed.

(b) The man the girl the boy met told jokes laughed.

MA also accounts for the preference for analysing a clause as main, rather than subordinate, which makes (a) less complex than (b) below:

(a) The girls pushed through the door and ran into the yard.

(b) The girls pushed through the door fell all over each other.

as well as the preference for analyzing 'that'-clauses as complements rather than relatives, making (a) less complex than (b) in the following pair:

(a) It amazed the teachers that I knew the answer.

(b) It amazed the teachers that I had in high school.

This description of MA as a preference for minimizing the number of nodes that must be added to attach a new word to the current phrase marker has been criticized for being too dependent on assumptions about the underlying grammar, but similar results can be achieved by ordering the way attachments are considered (Wanner, 1980) (but this then requires an explanation for the ordering) or by trying to minimize the time it takes for a particular attachment possibility to be recognized (Fodor and Frazier, 1980; Cottrell, 1988). The latter description, in fact, provides the most general statement of MA, namely that the attachment that can be computed most quickly is the one that will be favoured; assuring that favoured attachments are the simplest to compute and represent will be the job of the grammar. For example, Fodor and Frazier (1980) suggest that grammar rules might be

> accessed in parallel and selected in terms of the outcome of a 'race' — the first rule or combination of rules that successfully relates the current lexical item to the phrase marker dominates subsequent processing. (Fodor and Frazier 1980: 434)

This statement might easily provide a good explanation for either the node- or the time-counting version of MA, but it should be noted that its authors actually presume a grammar like the one given earlier, so more rules implies more nodes, because they wish to present MA as a natural result of the architecture of their parser rather than as a heuristic strategy. Regardless of whether it is preferable to describe MA as a node- or time-minimizing phenomenon, however, it is agreed that MA should at least serve as a general guideline for processing.

Revision as Last Resort

Revision as Last Resort (RALR) is a principle that can best be paraphrased by saying that the HSPM is not crazy. In particular, RALR means that once a syntactic structure has been built, it is not to be taken apart and rebuilt unless information from subsequent words force it to be, for example because of "semantic incoherence", "pragmatic implausibility", or "syntactic-ill-formedness" (Fodor and Frazier, 1980). The assumption here is that syntactic structure is built as each word is input (a claim I will defend later). RALR is in contrast with "strict determinism" as described in Marcus (1980), because it does not require that syntactic structures *never* be undone. ATNs that do a depth-first search through their network and backtrack only when they get stuck, as most do, are following a RALR strategy. As an example, assuming the grammar given earlier, RALR would support the attachment of "for Susan" directly to the VP node in **(11)**, even without invoking MA again at this point. RALR supports this attachment because initially the NP "the book" would be attached directly to the VP node by MA, and to modify this NP with the prepositional phrase "for Susan" would require inserting an extra NP node in between the VP node and the existing NP node. Now, one could see this attachment being revised if there failed to be a unique referent for "the book", and if "for Susan" successfully resolved the reference problem. Not every wrong attachment will be as easily revisable, however. In some cases it may not be clear exactly which attachment was incorrect, or what set of grammar rules can be used to incorporate both the phrase found to be wrongly attached and the new input. In these situations, when even RALR fails, we say the HSPM has been "garden-pathed". Since the HSPM must be efficient enough to handle frequently arriving new words, while using only its limited memory resources, RALR is the strategy we would most expect it to have.

Verb-Frame Preferences

Another way the HSPM might simplify the disambiguation problem is by using information about the preferred arguments of the verb. These preferences, called "verb-frame preferences" (Connine et al., 1984), influence the preferred reading of sentences. For example, in **(12)** the fact that "want" usually does not take a PP complement (or more specifically a locative) may create the preference for having "on that rack" modify "the dress", whereas the fact that "positioned" often does have a locative may create the preference in **(13)** for interpreting "on that rack" as an argument to the verb (Ford, Bresnan and Kaplan, 1982).

(12) The woman wanted the dress on that rack.

(13) The woman positioned the dress on that rack.

Although it is generally agreed that this information is used very early in processing to guide the initial analysis of sentence (Fodor, 1978; Ford, Bresnan and Kaplan, 1982; Clifton, Frazier and Connine, 1984; Mitchell and Holmes, 1985), definitive evidence is hard to come by. Some of the earliest tests, done by Ford, et al. (1982), consisted of simply asking their subjects to identify their initial reading of a syntactically ambiguous sentence and were not even statistically analyzed, although their results generally concur with later experiments. For example, more recently, there have been timing studies that have shown that a mismatch between the present arguments and the preferred argument structure of the verb slows processing at the point at which the mismatch is detected (Clifton, Frazier and Connine, 1984; Mitchell and Holmes, 1985). These tests appear to support the early use of verb-frame preferences; however, the existence of other structural, and perhaps semantic, preferences make even timing studies slightly equivocal. For example, in an informal (and untimed) study, Mitchell and Holmes (1985) found that they could manipulate preferred structure by manipulating the semantics; they found that 81% of their subjects took "the church bell" as the direct object in **(14)**, compared with 34% in **(15)**.

(14) The villagers heard the church bell...

(15) The villagers read the church bell...

They also found that this preference was not simply result of "heard" prefering a direct object more than "read", as demonstrated by the fact that 22% of their subjects took "the newspaper" to be the direct object in **(16)**, compared with 97% in **(17)**.

(16) The villagers heard the newspaper...

(17) The villagers read the newspaper...

Clifton et al. (1984) argue that verb-frame information is used before any semantic or pragmatic integration is possible, although apparently pragmatic information may quickly override lexical preferences if there is a conflict. Moreover, none of these experiments resolve the question of exactly how much information is used. Verbs may expect particular linguistic phrase structures such as NP, PP, or complement clause; or, they may select thematic roles such as direct object, locative, or complement. Verbs may also select some feature or combination of features for their arguments, such as animacy, and the presence or absence of a feature on one argument may change the expectations for the following arguments, which leaves open the possibility that some verb-frame information may be prestored while some may be computed online.

1.3.3 Parallel Processing of Syntax and Semantics

The third and final major assumption that this research makes with respect to the HSPM is that sentence processing involves many subtasks, such as syntactic and semantic processing, that proceed in parallel. In particular, syntactic and semantic structures are computed for each constituent as it is heard (Marslen-Wilson, 1975; Marslen-Wilson and Tyler, 1980; Marslen-Wilson, Tyler and Seidenberg, 1978), and not derived from each other. (See Von Eckardt and Potter (1985) for experimental evidence against the view that semantic representations are just recodings of completed syntactic representations.) Possible degrees of interrelation between these processes range from there being no separate semantic processing, a view that no one takes, to there being no separate syntactic processing as suggested by Small:

> The theory of syntax is an artifact that cannot be used as a foundation for parsing; stereotypic patterns of lexical interactions cannot account for the highly particular nature of context and usage (Small, 1980: 12).

which is a view that a few may take (Riesbeck and Schank, 1978; Small, 1980) but far more refute (Marslen-Wilson and Tyler, 1980; Frazier and Rayner, 1982; Rayner, Carlson and Frazier, 1983). Just considering views that incorporate both syntactic and semantic processing, then, one possibility is that the syntactic process tries not to combine any structures until syntax or semantics resolves any ambiguities (Marcus, 1980). If disambiguating information fails to arrive before memory capacity is exhausted, then supposedly the parser must make a guess and garden paths result. This view is not very appealing, because people do not appear to delay decisions in this way (Frazier, 1978; Frazier and Fodor, 1978; Marslen-Wilson and Tyler, 1980; Hirst, 1984; Frazier and Rayner, 1982; Rayner, Carlson and Frazier, 1983). Furthermore, the HSPM often garden-paths without exhausting memory capacity (Milne, 1982).

Another view of the interaction of syntax and semantics is that the HSPM builds all the syntactic possibilities in parallel and then asks the semantic process to choose one. Crain and Steedman (1985) declare that in instances of ambiguity, the interpretation that is more plausible, or that refers to an entity already established in the hearer's mental model of the domain of discourse, or that carries fewer unsatisfied, but consistent presuppositions or entailments will be preferred to an interpretation that is less plausible, fails to have a referent, or carries more unsatisfied presuppositions or entailments. They present experimental evidence based on grammaticality judgements showing how context can be used to create and to prevent garden paths and how structural preferences apparently can be eliminated by equalizing plausibility, referential success, and unsatisfied presuppositions and entailments. For example, they found that priming eliminates the garden path effect in **(18)**:

(18) There were two horses being raced, one out in the field and the other past the barn. The horse raced past the barn fell.

and semantics resolves the reduced-relative versus active-verb ambiguity in **(19)**, but supposedly garden paths **(20)**.

(19) The cheater furnished the answers passed the test.

(20) The genius furnished the answers passed the test.

Similarly it may be that pragmatics (here taken to be a semantic process), can also influence interpretations of sentences. (This claim has been argued by Moyne (1985).) For example, in **(21)** the PP "about his rude remarks" modifies the VP "was sorry" instead of the NP "an argument" perhaps because the pragmatics of being sorry makes it easy to assign it to a reason, whereas in **(22)** the PP "at 10:45 last night" is taken to modify the NP because the pragmatics of being sorry make assigning it an exact minute in time rather difficult.

(21) John was sorry after an argument about his rude remarks.

(22) John was sorry after an argument at 10:45 last night.

Manipulating pragmatic content is a bit tricky, however, because very often the way we change pragmatics is by changing the syntactic structure of sentences or the underlying case structures. For example, in **(23)** the PP "at 10:45 last night" modifies the VP "was shot" instead of the NP "a family argument".

(23) John was shot during a family argument at 10:45 last night.

(24) John was sad during a family argument at 10:45 last night.

This preference might exist because the pragmatics of shooting make it easy to assign it to a particular time, whereas in **(24)** the same PP is taken to modify the NP because the pragmatics of being sad make assigning it a time rather difficult, or the difference might exist because the passive VP "was shot" admits a modifier more readily than the copula-plus-adjective VP "was sad". Note that this hypothesis admits a bit of circularity because sentence generators may use a copula exactly when they want to express the fact that a predication was an ongoing phenomenon, not subject to assignment to a particular time. Also, even if these data suggest that semantic processes influence syntactic processes, they fail to prove when semantic information is used. Other researchers claim to have found that context does *not* always eliminate the effects of structural preferences (Rayner, Carlson and Frazier, 1983; Holmes, 1984; Ferreira and Clifton, 1986).[2] Also, Ford, et al. (1982) found that their subjects experienced the same attachment preferences for sentences like **(25)** and **(26)**, which have pronouns as lexical subjects and objects, as they experienced for sentences **(12)** and **(13)** in which lexical content words occured in place of the pronouns: namely, the preference in **(25)** for having "there" modify "everything", and the preference in **(26)** for having "there" modify "positioned".

(25) They wanted everything there.

(26) They positioned everything there.

Thus, people exhibit some consistent preferences for the resolution of attachment ambiguities, even in the absence of contextual information.

Another good explanation for why priming appears to eliminate some garden paths would be that the syntactic process tries to build an interpretation in its own preferred way, and if it detects an ungrammaticality, *e.g.* it finds a constituent it does not know how to use, it calls upon the semantic process to suggest a possible reanalysis of the earlier input (Ferreira, 1986; Ferreira and Clifton, 1986). Complex constituents already in focus because of priming are good candidates. Similarly, after semantic constituents are completed, the semantic process can check their plausibility, their referential success, and their satisfaction of presuppositions and entailments, and trigger a reanalysis if it detects a semantic error. The problem with grammaticality judgements is that they can only test subjects' eventual interpretation, and not how they reached that interpretation.

Thus, a third possible view of the interaction between syntax and semantics is that syntax computes structures independently, on the basis of its own preferences, but may use semantic information later to guide or trigger necessary revisions. To test this possibility, Ferreira (1986) and Ferreira and Clifton (1986) timed the eye movements of subjects reading minimally-attached and non-minimally-attached sentences, varying within sentence semantics and context. They found that even when subjects were primed for

---

[2] Some of these experiments have been seriously criticized, however. See Underwood (1985) for a discussion of problems in Rayner et al. (1983).

a non-minimally-attached sentence as in example (18) above, subjects spent more time reading the disambiguating information and the region immediately following it for non-minimally-attached sentences than they did for similar minimally-attached sentences. They argue that this proves that regardless of context, people still *initially* compute the syntactically-preferred structure and then reanalyze sentences that require non-preferred structures later. This view is appealing because it means that "only one analysis at a time is constructed and all incoming material is structured as it received" (Ferreira and Clifton (1986): 349). But, this view is also unappealing because it means that the HSPM may waste time creating structures it will later discard, just to make MA sentences run a little faster. Moreover, there is evidence that semantic information interacts with syntactic information in such basic tasks such as word recognition and fixation on pronouns (see Underwood (1985) and references therein), and in the within-sentence chunking of words (see Klatsky (1980:97) and references therein). Furthermore, there are many other reasons that subjects may spend more time reading the disambiguating material and the following segment in the non-minimally-attached sentences; for example, the parser may simply be responding to the hard work associated with building a relative clause representation, or suffering from a left-over "boggle effect" resulting from its recognizing that there is a semantic error but perhaps not knowing quite how to resolve it.[3] Moreover, in the primed experiments, reading disambiguating material for non-minimal-attachment sentences may take longer than for minimal-attachment sentences because this disambiguating material signals that the end of the complex noun phrase has been reached, which means the parser now has to integrate this noun phrase with context (Aaronson and Ferres, 1986), whereas the disambiguating material for minimal-attachment sentences just signals the continuation of the verb phrase.

1.3.4 Remaining Concerns

Although it is fairly easy to enumerate the major psychological issues, simply enumerating them does not reveal the whole story. In particular, the details with respect to how the various memory-limitation effects, structural preferences, and parallel-processing effects all interact have been sidestepped. Any complete account of these effects must also give an account of what happens when there are conflicts between principles. Moreover, it is difficult, if not impossible, to give an absolute account of these interactions because often it may not be clear which influences are at work, and the effect of a particular preference may depend on what other preferences are present. For this reason, most of the issues presented so far have been described as influential rather than rigidly constraining. Probably the best that any implementation of a theory can do is try to account for all known data and remain flexible enough to incorporate revisions to the theory triggered by any new data that are found. Thus, for example, a model might propose a means of rating preferences and then take the one with the highest rating as dominant or try to maximise the total rating by taking a combination of preferences. In the next chapter we will see how well previously proposed and implemented models account for the psychological issues that have been addressed here.

---

[3] Janet Dean Fodor, personal communication, June 1987.

Chapter 2

Review of Psychologically Motivated Systems

Although there have been a great many sentence-processing models proposed, only a very few are both computational in nature and psychologically plausible. Despite their relative rarity, however, existing psychologically motivated models illustrate that there are a variety of ways to account for memory limitations, structural preferences, and parallel processing of syntax and semantics.

2.1 Systems That Model Memory Limitations

One way memory limitations have been incorporated into sentence processing models is by limiting the number of constituents that the parser can see, either by forbidding it to look beyond a particular distance, or by forcing it to "forget" some information when it has used all available memory capacity. For example, the parser Parsifal (Marcus, 1980) includes a three-constituent buffer, a stack, and a set of rules that access only the buffer and a limited number of stack elements. These restrictions are meant to limit look-ahead and thereby model human garden-path behaviour. Church (1980), however, argues that to really model memory limitations Marcus's parser needs to have a bound on the depth of its stack. Church's system is a modified version of Parsifal, incorporating a limit on stack depth corresponding to the depth of allowable centre-embeddings. (So as not to exclude right- and left-embeddings, Church's parser also follows his "A-over-A early closure principle", which states that given two phrases in the same category, the higher closes when the next node parsed is not an immediate constituent of either, and the mother and all obligatory daughters have been attached to both nodes.)

Pulman (1986) takes a similar approach to implementing memory limitations in his modified shift-reduce parser. In particular, whenever the parser's stack grows beyond some fixed depth L and there are two projections of V (*i.e.*, either a VP or an S) adjacent to each other such that the one on top potentially contains everything needed by the one below, then he combines the two items. This combination, an operation called "Clear", composes the semantic information and discards the syntactic information describing the internal structure of the closed item. Supposedly this composition saves space because the semantic representation, although it resembles a complex hierarchical syntactic representation, is meant to describe a single semantic object. This approach to memory limitations is not altogether appealing, however, because it fails to account for the apparent fact that there is also a decrease in the availability of semantic or conceptual information about the first clause once the second clause has been processed (Von Eckardt and Potter, 1985).

Recall that one side-effect of memory limitations is that people can keep only a limited amount of material unstructured. Thus, some psychologically-based parsers have also tried to incorporate this limitation. For example, parsers described in Pulman (1986) and Schubert (1984) process each word as fully as possible as soon as it is encountered. In contrast, parsers derived from Marcus's orginal parser tend to be wasteful of memory in this respect because they delay attachment decisions for as long as possible, hoping that disambiguating information will arrive.

Parsers may also limit the number of ambiguities they choose to represent. A great many systems do not represent any ambiguities, but usually for reasons other than concern for modelling effects of limited memory. Schubert's (1984) system is an exception in that evidence of memory restrictions has motivated Schubert to create a parser that stores at most three ambiguous constituents, and that incorporates expectation potentials that decay with distance (which gives it something akin to local association).[1]

Church (1980) presents a partial solution to ambiguity resolution by incorporating an idea called "pseudo-attachment", in which syntactic structures are represented as directed acyclic graphs and any node that could be attached in two ways would have two mothers in its structure (see Marcus (1984) for a discussion of a similar proposal).These compact representations of ambiguity, however, while accounting for memory limitations, ignore the psychological evidence against delaying attachment decisions. (For a discussion see section 1.3.1 and references therein.).

In contrast, Pulman's parser (1986) does not even attempt to be psychologically realistic in this regard because it computes all possible parses.[2] Moreover, fixing this flaw is not a relatively simple matter of just adding a decision mechanism to kill off all but one hypothesis as soon as the ambiguity is detected, as suggested by Pulman. At the very least, this addition would also necessitate adding an extra decision principle to make sure that right-branching sentences are not misanalyzed as centre-embedded

---

[1] Supposedly examples like (i), that Schubert has found to *not* be garden paths, have convinced him that "Marcus (1980) was essentially correct in positing a three-phrase limit on successive ambiguous constituents" (Schubert 1984:250).

(i) The boy | got | fat spattered on his arm.

Unfortunately, Schubert does not elaborate on his argument, leaving it rather unclear. In particular, Schubert does not indicate how sentence (i) above is to be divided into successive ambiguous constituents, or what makes the sentence unambiguous where similar sentences such as (ii) are not. (The divisions marked are my own best interpretation of Schubert's remark.)

(ii) The boy | got | fat | spattered.

Originally, Marcus argued that, because of memory limitations, the HSPM stores at most three successive phrases, causing sentences such as (ii) to be garden paths.

[2] Gorrell's (1987) demonstration that competing structures are computed in parallel is not inconsistent with this criticism, because he argues that only one hypothesis is active at the attentional level.

ones, because the current implementation depends on the fact that overly-deep centre-embedded sentences will be killed because processing them requires using more than fixed depth of the stack.

2.2 Systems That Include Structural Preferences

Another way in which parsing models have tried to incorporate psychological reality is by trying to model observed structural preferences such as Right Association, Minimal Attachment, and verb-frame preferences.

2.2.1 Models of Right Association

Recall that the principle of Right Association (RA) essentially says that when building a parse tree, new words should be attached into the nearest enclosing phrase, *i.e.*, as low as possible along the right frontier of the tree.[3] The rationale behind this preference is that the HSPM is something like a stack, so that it wants to keep a phrase open as long as possible because once it goes back to a higher level (conceptually popping the lower level off the stack), it can never return to the lower level. Also, if all the input can be attached at the lowest level, then the parser can just stop there and not have to work its way back up to the top. ATNs can incorporate this stack description of RA rather literal-mindedly by scheduling (trying) all SEND arcs and JUMP arcs after arcs of every other type (Wanner, 1980). Thus, an ATN can be sure that all other choices have been exhausted before it returns to a higher level. This method can, and has, been adapted for use in deterministic parsers by making sure that phrases are closed only when the parser can neither do an attachment nor make a prediction about upcoming input (Church, 1980). Similarly, a shift-reduce parser can incorporate RA by resolving all conflicts between shift and reduce operations in favor of shifts (Shieber, 1983).

In addition to these models that intentionally follow RA, there are some models that, although they claim not to be motivated by structural preferences (and hence are not directly psychologically motivated), implicitly follow RA in their attempt to account for the RA data or in their desire to expedite processing. For example, Wilks et al.'s parser (1985), although explicitly intended to use only "semantic information" (without appealing to syntactic principles) to determine the attachment of PPs, implicitly incorporates RA to attach PPs because it always considers attachments in right-to-left order. Similarly, Pulman's system (1986) is compatible with RA because its "Combine" operation only combines entries on top of the stack with entries immediately below them (*i.e.*, it considers attachments in right-to-left order), although his system computes all other possible parses as well.

---

[3] Note that the nearest enclosing phrase may not always be the phrase containing the words immediately to the left of the current word. Rather, it may depend on the effects of other structural principles such as Minimal Attachment.

2.2.2 Models of Minimal Attachment

Minimal Attachment (MA) has also been incorporated into a variety of sentence-processing models. The node-counting version of MA requires that the parser should add the fewest possible non-terminal nodes to link a node with those already present. Given that for this statement of MA *the* minimal attachment is determined by the particular grammar assumed, we can only look at how different models implement MA using a particular grammar, for example the one given earlier:

S ← NP VP
VP ← V NP (PP)
NP ← Det N
NP ← NP PP

Once again, an ATN can incorporate MA by means of a general scheduling rule: try all CAT arcs and WORD arcs before trying any SEEK arcs (Wanner, 1980). This rule supports MA by making sure that the parser tries all attachment possibilities at the current level before SEEKing to a lower level, thus conserving computational resources that might be wasted during unnecessary shifts between levels. Wanner (1980) argues that the existence of MA may be explained by the following:

1. Minimal Attachment guarantees that the attachment requiring the fewest number of arcs will be tried first.
2. Minimal Attachment minimizes the number of SEEKs per parse.

However, this explanation fails to hold if the grammar does not include an asymmetry between the attachment of modifiers to NPs and the attachment of modifiers to VPs (see section 1.3.2). Note that in cases where MA and RA might appear to conflict given a partially completed parse tree, Wanner's fixed rules for scheduling arcs ensure that his ATN will always apply RA within MA; that is, RA never overrides MA. Unfortunately this also means that ATNs will not be able to easily incorporate any other challenges to MA, such as verb-frame preferences.[4] A deterministic parser can implement MA by always trying attachment rules before rules that predict new structures, although "attach before predicting" statements of MA all lack explanatory force because it is not always most efficient to attach first, and it may be that this ordering of operations explains MA rather than the other way around (Church, 1980). shift-reduce parsers can also incorporate MA for the above grammar by means of a general conflict-resolution rule: just resolve all conflicts between two reductions by taking the longer of the two (Shieber, 1983). Unfortunately, to apply this rule, shift-reduce parsers must delay making attachment decisions until phrases are completed, which is psychologically implausible. As a

---

[4] Moreover, ATNs have been criticised as psychological models of language understanding because they often totally disregard apparent memory limitations (Frazier and Fodor, 1978), and because there is some doubt about how ATNs' pushdown stores and HOLD mechanisms relate to concepts described in the memory literature (Levelt, 1978). There is also a belief that ATNs, which have the capability of a universal Turing machine (Woods, 1970), are just much too powerful; see Levelt (1978: 58–59) for a discussion.

third example, the parser described by Wilks et al. (1985), although it initially considers attachments from right to left, when it cannot find a noun or verb that prefers the object to be attached (Wilks et al. do not distinguish between the two cases), then attempts to find an attachment "using the case preferences of the preposition starting with the main sentence verb and moving rightwards" (p. 782). If this search also fails, then the parser attaches the PP to the main sentence verb. Together, these two strategies give MA within RA.

The timing account of MA, which essentially says that the attachment that is recognized first is preferred, is not as straightforward to program and so it has been used less frequently in computer implementations. Cottrell (1988) presents one such implementation:

> The model works by combining top-down expectations and bottom-up input. Imagine a grammar representation which is *active*, in the sense that as parts of productions are recognized, activation spreads to the next part of the production. Different productions in the grammar *compete* for the attachments of constituents that are found in the input. The more nodes involved in a particular interpretation, the farther the activation has to spread, and the longer it takes to activate those nodes that the input actually attaches to. Meanwhile, if there is a representation that matches the input that involves fewer nodes, this will become activated faster and get a head start over the representations involving more nodes. (p. 126)

Then, once a production has a head start, it can collect more support than its competitors because only it will be ready to attach incoming nodes. Finally, when a production is sufficiently better supported than its competitors, the lagging productions will be killed off, leaving only the minimal attachment. This formulation still requires that the grammar writer make sure that preferred structures involve fewer nodes, but it provides a more natural explanation of why having fewer nodes, and hence MA, is advantageous.

One possible compromise between these two views would be to associate a cost with each grammar rule and require that alternative attachments be compared according to their net goodness, *i.e.*, their goodness less their cost. This system would ensure that structures using fewer rules would generally be preferred to structures using more. Such a view has been proposed by Schubert (1984). Like the first approach, however, Schubert's approach has only as much explanatory value as its grammar and its method for accessing cost, which unfortunately Schubert does not describe.

2.2.3 Models of Verb-Frame Preferences

Systems that model verb-frame preferences use these preferences as heuristics to guide the initial analyses of sentences. None incorporate the preferences as a revision tool, but then, except for backtracking models such as ATNs, none even attempt to incorporate revision. The variance in the implementation of verb-frame preferences exists primarily in the degree of flexibility in the representation of these preferences. Obviously, the greater the resolution, the better the chance of getting an accurate account of these preferences if they are ranked correctly, but the more difficult it becomes to determine a correct ranking. Shieber (1983) describes one possible implementation of verb-frame preferences. First, Shieber organizes verbs into several broad classes by defining different "preterminals" to distinguish different syntactic argument structures. Thus, for example:

V0 requires no arguments
V1 requires NP
V2 requires NP PP
V3 requires infinitive argument
V4 requires adjective
V5 requires PP

Then, he specifies for each preterminal–word pair whether or not this usage of the word is strong or weak. So, to encode the fact that "want" usually has only an NP object, he would make the pair (V1,"want") strong and all other pairings of "want" and a verb-form preterminal weak. The parser then incorporates these preferences by resolving conflicts between two possible reductions in favor of the one that has the strongest leftmost stack element and then using other preference information to handle any remaining conflicts. This system is not very flexible because it only provides two possible preference ratings ("weak" and "strong") and is also limited because it does not use any information about the argument structure other than the syntactic category. However, it is fairly easy to verify (or refute) that the information that one is able to encode is in fact correct.

In contrast, Wilks et al. (1985) encode the preferred argument structures of nouns, verbs, and prepositions, including information about syntactic categories as well as thematic roles. Thus, for each noun and verb, they have a list of case roles, prepositions that might flag those cases, and selectional restrictions for the case fillers. So, for example, "position" has a locative case, flagged by "at", "in", "on", or "by", that must be filled by a place. For each preposition, there is a list of words it might modify, cases it might flag, and types of objects it might have. Once again, however, preferences are binary (either they exist or they do not) so there is not much flexibility. In particular, preferences are not ranked so there is no way to allow for differences in the strengths between preferences. Recall that Wilks et al. consider attachments serially from right to left. Schubert (1986) points out that this means that they will misparse either (27) or (28) because the correct reading of (27) (with "to London" attached to "ticket") requires a preference for having a destination case for "ticket", while the correct reading of (28) (with "to London" attached to "mailed") requires a preference for having a destination case for "mailed", but in (28) the existence of a destination case for "ticket" would be discovered first, leading to attachment, even though 28 of 29 of Schubert's respondents attached "to London" to "mailed".

(27) Joe lost the ticket to London.

(28) Joe mailed the ticket to London.

In order to avoid this problem, Schubert encodes information about the arguments of verbs by means of "maximal expectation potentials" assigned to verbs individually. Thus, a verb that strongly expects a certain type of argument will be given a relatively high maximal-expectation-potential for that argument. Unfortunately, Schubert does not describe how these potentials are determined, just that "The maximal expectation potentials of the daughters of a node are fixed parameters of the rule instantiated by the node"(p.249). To fully evaluate this proposal it would be necessary to know how these potentials are to be determined and whether these potentials are specified in terms of syntactic category or thematic role. Because of its resolution, this system allows comparisons between the strengths of argument structures; however, assigning maximal potentials correctly will require much more effort than a less flexible system and it is yet to be seen whether a consistent assignment is even possible.

2.3 Systems That Compute Syntax and Semantics in Parallel

In the interest of modularity and simplicity of implementation, most computational models of natural language understanding have been content to do semantic processing after syntactic processing has been completed, either for the entire sentence or for completed constituents (see Hirst (1983; 1987) and Pulman (1986) for discussions). There are some systems that build semantic interpretations incrementally on a word-by-word basis, however.

Absity is one such system (Hirst, 1983). The syntactic processing is done by means of a deterministic parser, Paragram (Charniak, 1983), modeled after that of Marcus (1980). The semantic processing, inspired by Montague semantics (Montague, 1973; Dowty, Wall and Peters, 1981), proceeds in tandem with the syntactic processing. In particular, for each attachment rule, there is a semantic counterpart; so when an attachment is identified, both the corresponding syntactic and semantic rules are executed. Semantics can interact with syntax by means of the Semantic Enquiry Desk that responds to queries about which of a set of operations would be semantically preferred.

The Semantic Enquiry Desk determines the attachment it prefers by following a fixed schedule of attachment strategies. For example, to handle PP attachment questions, after verifying that all selectional restrictions have been satisfied (usually the word-disambiguation process filters selectional mismatches), it prefers the first available of:

1. any attachment that would create a description of an entity already in the knowledge base (*i.e.,* achieves *referential success*),
2. a unique attachment that describes an entity similar to ones already in the knowledge base, (*i.e.,* is *plausible*),
3. any attachment that would satisfy expectations of the verb,
4. the rightmost plausible attachment to an NP
5. if there are more verb cases left, any attachment to an NP that does not fail referentially or, if all else fails, to the verb (minimal attachment),

6. else if it is working on the last verb-case, any plausible attachment to the verb, any attachment to an NP that does not fail referentially, or, if all else fails, the rightmost such attachment.

This general algorithm is meant to combine lexical (verb-frame) preferences as described in Ford, et al. (1982) and the semantic preferences described in Crain and Steedman (1985) (see section 1.3.3 for a discussion). The operation of the Semantic Enquiry Desk is wholly independent from the syntactic component, so the parser can respond to semantic preferences, but semantic failures need not trigger syntactic failures. Words are also disambiguated by a separate, independent process.

Pulman (1986) also incorporates Montague-inspired semantic processing in his parser. However, in Pulman's parser, syntax acts as a control structure for building semantic structures. Thus for each syntactic rule there is a semantic rule, but only the semantic rules produce representations.

In Cottrell's connectionist parser (1988), syntactic and semantic representations are computed in parallel by independent processes that may interact asynchronously. Although he has not yet linked these two processors, he proposes to do so by means of binding nodes so that the pattern of activation in one network can influence, by means of support, the activation in the other network. Since the two processes are completely independent, it may be the case that one runs faster than the other, so it is possible that a syntactic interpretation might succeed on the basis of semantic strength alone or vice versa. Cottrell suggests that this might be a source of garden-path sentences. In general, however, it is expected that one network influences the other only when preferences within that network are too weak to resolve an ambiguity on their own.

Huang and Guthrie (1985) describe a system based on Semantic Definite Clause Grammars (see Huang (1985)) that attempts to satisfy syntactic and semantic constraints in parallel. The system, implemented in Prolog, attempts to satisfy a conjunction of syntactic and semantic predicates which include a top-down description of the syntactic structure and a set of semantic constraints on compatibility between such things as adjectives and nouns, subject NPs and verbs, and verbs and object NPs (compatibility to be determined by selectional restrictions). The system produces all parses not blocked by unsatisfiable predicates, and requires an "unpredictable" amount of backtracking, making it entirely implausible from a psychological standpoint.

2.4 The Frazier and Fodor Proposal

Frazier and Fodor (1978) propose a sentence parsing model they call the "Sausage Machine" (SM) that accounts for many of the psychological issues that I have described in this thesis thus far, appears to allow for the incorporation of issues it does not account for directly, and introduces some ideas that appear promising from a pragmatic standpoint. The SM is a two-stage device in which the first stage is limited to looking at the next five or six words of input and building a single structured representation of them. As the first stage, or "Preliminary Phrase Packager" (PPP), completes these representations, it passes them on to the second stage, forgetting what it has done as it begins to build a structure for the next set of words. Meanwhile, the second stage called the "Sentence

Structure Supervisor" (SSS), takes the packages it receives from the PPP and combines them into a complete parse tree. As a consequence of its limited view, the PPP usually only forms representations of phrases, clauses, or partial clauses, but may complete a structure for a whole sentence if the sentence is short enough. The SSS, unlike the PPP, can survey the whole parse tree as it is built, keeping track of long-distance dependencies and structural commitments. The motivation for this model lies in the fact that although working memory appears to be very limited in the number of units it can store at once, it can apparently store more information in its limited space by "chunking" it or storing it as structured units. This model differs from earlier two-stage parsing models (see Frazier and Fodor 1978: 292, for a discussion) in that the units produced by its first stage are defined by their size rather than their syntactic shape, usually with limited knowledge of the overall structure of the sentence as it is being built.

[5] Within each stage, processing proceeds in a combined bottom-up and top-down manner, and is governed by the principles of RA, MA, and RALR. Thus, the SM model incorporates memory constraints, structural preferences, and a degree of parallelism, making it particularly interesting from both a psychological and a computational point of view.

2.4.1 The Sausage Machine's Psychological Appeal

The SM model is appealing from a psychological point of view because it gives a good account of both sentence complexity and structural preferences.

SM Account of Sentence Complexity

According to Frazier and Fodor (1978) (FF), sentences that require distant attachments are more difficult to process or "more complex" than sentences whose attachments are all local, and they show how this can best be explained by the SM's interpretation of the effect of limited memory. One possible interpretation of the effect of memory constraints would be that as the parser tries to hold more and more nodes simultaneously, the mere strain of having to hold so many nodes, and perhaps apply compression and compaction operations, causes sentences to seem complex. This explanation cannot be completely correct, however, because then, for a sentence like **(29)**, either the HSPM couldn't keep the top level S-node in memory long enough to attach "yesterday" to "said", or if it could, then the reading that attaches "yesterday" to "said" and the reading that attaches "yesterday" to "had taken" should be equally complex because they both require the same number of nodes to be stored in memory at the same time.

(29) Tom said that Bill had taken the cleaning out yesterday.

However, the existence of sentence **(30)**, which is superficially similar to the sentence **(29)** except that it requires the more distant attachment and yet is more complex than **(29)**, dispels both these possibilities.

---

[5] This use of the term *two-stage* is thus also different from that of Weinberg (1987), who describes what might also be called a *two-pass* system: the first pass builds a parse tree, the second establishes binding relations.

(30) Tom said that Bill will take the cleaning out yesterday.

Note that unless you explicitly think of **(30)** as the answer to a question like "What did Tom say about Bill yesterday?", it seems to be only marginally acceptable as a sentence. The SM explanation for why limited memory causes some sentences to seem more complex than others is that these constraints cause the parser to err as it tries to abide by them. In other words, because the parser "knows" that not all the nodes can be stored simultaneously, it may try to make decisions that are locally correct, but may be wrong with respect to the whole sentence. Then it may be left with an inconsistent interpretation or it may be forced to reanalyze and revise its interpretation. By forcing the PPP to shunt packages of words to the SSS before the PPP memory is overloaded, the SM supports this explanation of sentence complexity. Moreover, FF show that there are discontinuities in the complexity of sentences that support the existence of two separate stages. For example, the complexity of high and distant attachments supposedly remains constant rather than increasing with height or distance, and longer phrases or clauses that tend to be packaged by themselves, hence limiting errors, apparently can be attached distantly without increasing complexity. This corresponds well with the proposal that the SSS be allowed to view the entire sentence in structured units.

The SM model also gives a good acount of the difficulty caused by centre-embedded sentences. The SM explanation for centre-embedded sentences is that their correct analysis requires that the PPP not group adjacent phrases together as it is designed to do (Frazier and Fodor, 1978). For example, the first six words of sentence **(31)** will be misanalyzed by the PPP as a conjoined noun phrase, because of MA.

(31) The woman the man the girl loved met died.

Blumenthal (1966) showed that people make similar mistakes in processing centre-embedded sentences. Moreover, the complexity of centre-embedded sentences appears to be reduced in cases where the lengths of constituents would help the PPP make the correct packaging decisions and then leave the SSS to resolve the embedding, as in **(32)** (Frazier and Fodor, 1978).

(32) The very beautiful young woman the man the girl loved met on a cruise ship in Maine died of cholera in 1962.

In **(32)** the length of the initial noun phrase would lead the PPP to package it as a unit in itself; then,"the man the girl loved met" would form the next package, which the PPP would recognize as embedded. Similarly, the next two packages: "on a cruise ship in Maine" and "died of cholera in 1962" are correct with respect to the embedding. In contrast, sentence **(33)** would mislead the PPP, because it would prefer to combine the first two NPs "the woman the sad and lonely old man" into a single, conjoined noun-phrase and "the pretty little schoolgirl loved" into an active clause, leaving it with an uninterpretable chunk "with all her heart met died".

(33) The woman the sad and lonely old man the pretty little schoolgirl loved with all her heart met died.

Centre-embedding difficulty cannot be just a problem in storing a great many nodes; otherwise sentence **(34)** and sentence **(35)** would have nearly the same complexity since the second sentence is only marginally longer, but the second sentence is in fact substantially more difficult (Frazier and Fodor, 1978).

(34) I saw a boy who dropped the delicate model airplane he had so carefully been making at school.

(35) I saw a boy who dropped the delicate model airplane he had so carefully been making at school into a puddle cry.

Moreover centre-embedding difficulty cannot just be because "the human parsing mechanism is fundamentally incapable of operating recursively"(Pulman 1986: 214), for we can actually reduce the complexity of the sentence **(35)** by merely increasing the length of the last two constituents as in **(36)**, thus preventing them from being attached together in a single package (Frazier and Fodor, 1978). (Assuming that packages are around six words long, package boundaries for **(36)** would most likely occur at the places marked with "|".)

(36) I saw a boy who dropped | the delicate model airplane he | had so carefully been making | at school into the puddle | of mud beside the back door | reach down and pick it up | by its broken tail fin.

Similarly, if we assume that a pause forces a package break, we can see how adding pauses at "|" makes **(37)** easier to understand.

(37) The woman | the man the girl loved met | died.

Note that increasing the length of selected constituents or adding punctuation will not trigger Pulman's "Clear" operation (and perhaps give him space to resolve the embedding) as it might for a right-recursive structure because his "Clear" only operates on adjacent projections of V (*i.e.,* VP or S). However, Pulman may be partly correct in his claim about the difficulty of recursive structures since obviously **(38)** is still difficult even though the presence of the complementizers should make analyzing the structure easier.

(38) Women that men that girls met loved died.

Nevertheless, the existence of this example is not conclusive evidence, because in English "that" can also be used as a determiner and the initial segmenter might not check for number agreement. Thus, the difficulty of centre-embedded sentences appears to come in large part from the difficulty in segmenting them properly, strengthening the claim for two-stage parsing.

SM Account of Structural Preferences

In addition to explaining sentential complexity, the SM model also accounts for several structural preferences such as Local Association (LA), Right Association (RA), Minimal Attachment (MA), and Revision as Last Resort (RALR). In the SM, the limited view of the PPP causes LA because non-local attachments are not seen by the PPP and so they are never considered. MA, RA, and RALR, on the other hand, are not strictly required by the SM model, but would naturally be expected by it because they make the SM more efficient in a psychologically plausible way. In particular, following MA and RALR means that the SM will access a minimum number of rules and build a minimum number of structures (Fodor and Frazier, 1980). RA reduces shifts between constituents in the SM, and tends to increase the size of PPP packages, reducing the strain on the SSS and giving it more "time" to look for different attachment possibilities (Fodor and Frazier, 1980). To achieve RA, Fodor and Frazier (1980) suggest that the SM could just

> scan the phrase marker and attend to its dangling nodes in sequence as it moves around the bottom of the structure (p. 455)

As we have seen, however, there are sentences such as **(11)** where MA dominates RA, and sentences such as **(2)** where RA appears to violate MA, and sentences such as **(7)** where both RA and MA appear to be violated. Fodor and Frazier claim that these violations occur because although the PPP always abides by RA subject to MA and RALR, it does so only within its limited viewing window, and thus may unknowingly create an attachment which violates RA, MA, or RALR with respect to the whole parse tree. This explanation may be partly correct, although other factors may be at work as well; consider the sentence:

(39) Joe included my gift for Susan.

In **(39)**, RA overrides MA and RALR without the involvement of any viewing limitations. One possibility is that the preference here is determined by the relative strengths of the lexical forms of "include" and "include for" (Ford, Bresnan and Kaplan, 1982). However, verb frame preferences cannot completely displace the SM account of RA and MA as evidenced by **(40)**

(40) Joe took the book that I had wanted to include in my birthday gift for Susan.

in which "for Susan" supposedly associates with "my birthday gift", in contrast to **(41)** in which "for Susan" would associate with "took" (Fodor and Frazier, 1980).

(41) Joe took the gift for Susan.

Consequently, the importance of the SM account of structural complexity cannot be ignored.

2.4.2 The Sausage Machine's Computational Appeal

The SM model is also appealing from a computational point of view because it attempts to make the best use of time and information available to it. Firstly, the SM benefits from having two stages running in parallel. This allows it to do simple segmentations while it is also resolving major attachments, locating fillers for gaps, and perhaps revising structures on the basis of semantic information. Also, although the SM is described as primarily a bottom-up device, allowing incoming words to trigger appropriate grammar rules, FF also propose that it should have some top-down, expectation-driven capabilities. FF claim (but do not provide evidence) that people seem to "anticipate predictable aspects of the phrase marker", although efficiency concerns alone would seem to warrant making the SM fully information-driven. Hence, FF

> propose to permit both the PPP and the SSS to postulate obligatory nodes in the phrase marker as soon as they become predictable, even if their lexical realizations have not yet been received. The PPP, for example, upon receiving a preposition or an obligatorily transitive verb can enter an NP node as its right sister before any elements of the noun phrase have been located in the lexical string (Frazier and Fodor, 1978: 316).

Similarly, the SSS can add nodes to keep track of some obligatorily parallel constructions. These predictions will then help the parser spot gaps and errors of omission, as well as resolve some temporary ambiguities. The strength of this mixed type of processing lies in the fact that the parser is able to take advantage of the information it has collected, without ever being forced to make predictions before it has accumulated sufficient evidence. Thus, the SM should not need to retract a hypothesis nearly as often as a purely top-down system.

2.4.3 Problems with the Sausage Machine

Despite its psychological and computational appeal, the SM model as originally described has at least one obvious flaw and several critical omissions. (Also see Wanner's (1980) general critique.) For example, the SM's account of MA is much too dependent on ad hoc details of the grammar that Frazier and Fodor (1978) have assumed that the SM would use. This particular flaw has been criticized elsewhere (Church, 1980; Cottrell, 1988; Schubert, 1984; Hirst, 1987), but it has also been noted that there is a fairly simple way around the problem. In Fodor and Frazier (1980), the authors suggest that the parser consider alternative parses in parallel and that the first one that is completed will be the one that "dominates subsequent processing". Since the path with the fewest nodes will probably be completed first, it is equivalent to say that the minimal attachment is the one that the parser finishes building first. This statement of MA thus reduces MA to a general processing heuristic, to be explained by the HSPM's apparent need to be efficient rather than by some ad hoc set of grammar rules. This statement of MA also requires that that the grammar or the parser be structured so that preferred structures take less time to complete than other structures. More disturbing than any flaws the SM may contain, however, are the many issues the SM simply fails to address. The model as originally presented says nothing about how the two stages might communicate with each other, how verb frame information might be used, how revision would work, and, most seriously, how semantics and pragmatics might fit in. As argued earlier, any realistic parsing model must account for these issues, and thus any interesting implementation of the SM must first rectify its omissions. This thesis will address some of these issues.

[illegible]

[illegible]

[illegible]

[illegible]

Chapter 3

A Race Model of Parsing

The processing models described so far account for only a subset of the important psycholinguistic data outlined in the first chapter, without making clear how accounts of the remaining data could be integrated. This leaves open the possibility that they will not be able to integrate the remaining data consistently. What is needed, and what this thesis proposes, is more of a "kitchen sink" approach to sentence processing. In particular, it proposes a relatively robust and psychologically plausible account of how the various psycholinguistic factors described earlier might be incorporated and integrated.

3.1 Overview

The general framework of the sentence processor being proposed here is a two-stage parsing device inspired by the Sausage Machine (SM) (Frazier and Fodor, 1978; Fodor and Frazier, 1980). The first stage incorporates each word, as it is received, into the current fragment of syntactic representation. These fragments are limited in size so as to respect the limits on the capacity of working memory (see section 1.3.1 for a discussion). When a fragment approaches the capacity of working memory, the first stage passes it to the second-stage processor and begins a new fragment. The second-stage processor, running in parallel with the first stage, then incorporates each fragment it receives into a syntactic representation for the entire sentence.

The proposed parsing model extends the SM model in that it makes explicit claims about the interaction between syntactic and semantic processing. In particular, semantic interpretations are to be computed in parallel with syntactic structures incrementally, on a word-by-word basis (see section 2.3). Syntactic structures are derived independently by a separate syntactic processor, allowing semantic interaction only at selected points, such as when needed to resolve an ambiguity in attaching a phrase or a clause. The parser also incorporates a mechanism for resolving lexical ambiguities in parallel.

The proposed parsing model also attempts to remain independent of ad hoc aspects of its grammar, *i.e.,* aspects not constrained by the grammatical theory, and hence incorporates only a generalized statement of Minimal Attachment.

The grammar used for implementation is derived from Government-Binding theory (GB) (Chomsky, 1981; Abney, 1987; Cowper, 1987) and incorporates Abney's hypothesis "that Det selects a projection of N, not vice versa"(Abney 1987:74) (see sections 3.2.2

and 4.2).[1] The model differs from previously implemented GB parsers in that it attempts to integrate grammatical constraints and performance constraints. Other GB models have focused on grammatical constraints, ignoring most of the psychological issues we have discussed here. For example, Wehrli (1984) presents a "chart" parser intended to compute all possible structures for a given input sentence. The parser does not process sentences from left to right, but instead first assigns lexical categories to all the words in its input sentence. It then locates the lexical heads and parses their specifiers using a deterministic transition mechanism working from the head backward. Next, it attempts to combine constituents with preceding constituents either as a complement, an adjunct, or as part of one of two phrase structure rules, S → NP VP and $\overline{\text{S}} \rightarrow$ Comp S. Then finally, it invokes several modules that check for well-formedness conditions. Dorr (1987) presents another GB parser, but it builds only skeletal syntactic representations using a purely top-down algorithm that makes predictions based on "underspecified phrase-structure templates" and information about semantic roles from the lexical entries, and does not appear to address any ambiguity problems. Abney and Cole (1985), conversely, describe a GB parser that builds syntactic structures using a purely bottom-up algorithm in which structures may attach themselves to neighbouring structures if and only if the two constituents are related by means of predication, functional selection, or θ-assignment (a semantic role), however the system ignores problems of lexical ambiguity and uses a simplistic attachment rule to resolve structural ambiguity; namely, it first tries all the verbal (minimal) attachments in right to left order regardless of distance, preference, or priming, and, if necessary, then tries all noun phrase attachments in left to right order.

3.2 Accounting for the Psychological Issues

As promised, the sentence processor proposed here accounts for memory constraints, structural preferences, and the interaction between the syntactic and the semantic processors.

3.2.1 Memory Constraints

The sentence processor's account of memory constraints is identical to that of the SM proposal. As in the original SM model, the first stage of the parser has a capacity of approximately six words and builds a structure for these words without benefit of remembering previous structures it has constructed, thus incorporating a strict account of the memory limitation data. Similarly, the memory space of the second parsing stage is also limited in size, but because it processes larger units, it has the capacity to represent entire sentences. This corresponds to the fact that structured information requires less capacity than unstructured information. (See section 1.3.1 and references therein.) This account of memory constraints provides a good explanation for Local Association, because non-local attachments will be beyond the view of the first stage,

---

[1] From now on we will use the term "determiner phrase" (DP) to describe the traditional noun phrase and reserve the term "noun phrase" for describing the phrase structure headed by a noun.

and also provides a good explanation for sentence complexity (see section 2.4.1).

3.2.2 Structural Preferences

The original description of the SM model proposed that certain principles would guide the parser in resolving attachment ambiguities, but essentially left open the design of the decision mechanisms other than that they be governed by those principles in some natural way. Moreover, in our discussion of the psychological data that any model of the HSPM must account for, we identified several issues that the SM failed to account for, namely verb-frame preferences and syntactic-semantic interaction. Thus, my model attempts to more clearly and more fully explicate these issues. In particular, the parser builds phrase structures by projecting them from lexical heads according to the lexicon and $\overline{X}$-theory.[2] Phrasal attachments are determined according to expectations created by selectional information stored in the lexicon. In cases where the grammar permits more than one attachment of a particular word or fragment, the attachment that can be computed and verified most quickly is chosen. This approach to disambiguation is intended to reflect the fact that the underlying neural hardware is a highly parallel device, in which information, presumably including competing sentence structures, is stored by means of groups of neurons that may be activated at different times and rates. Thus, the resulting preferred reading for a sentence would be determined by the timing relationships between the activation of these representations.[3] In a predominantly serial model, one can simulate this behaviour by introducing a limited (and relatively easy to control) degree of parallelism.

Given two structures it wishes to combine, my parser starts a processing module for each possible method of justifying an attachment (that is for each lexical, structural, and semantic preference), ranks their results according to the relative time they took

---

[2] The reader is directed to *Lectures on Contemporary Syntactic Theories* (Sells, 1985) or *An Introduction to Syntactic Theory: The Government-Binding Approach* (Cowper, 1987) for a gentle introduction to $\overline{X}$-theory and Chomsky's theory of Government and Binding.

To briefly summarize the relevant aspects of $\overline{X}$-theory: Words belong to lexical categories such as Determiner (Det), Noun (N), Verb (V), Adjective (Adj), Adverb (Adv), or Preposition (P). Phrase categories include NP, VP, and PP. Each instance of a particular phrase category contains an obligatory element which is called the head of the phrase. (Usually phrase categories are named after their heads, so for example, the head of an NP is a noun.) The phrase category is then considered to be a "projection" of its head. Functional arguments and modifiers of the head introduce intermediate levels of structure. We distinguish different levels of structure using "bars", for example, we have N, N', and N'' (NP). These intermediate levels are also called projections, so the phrase category is usually called the "maximal projection" of the head. $\overline{X}$-theory then requires (Cowper 1987:75) that:

(a) Every X (lexical category) must be the head of an X^{max} (its maximal projection).

(b) Every X^{max} must have a head of category X.

(c) Every non-maximal X^n must be dominated by X^{n+1} except X', which may be dominated by X'.

(d) X is left (right) peripheral in X'.

(e) X is left (right) peripheral in X''.

[3] For a similarly motivated approach to parsing, see section 2.2.2 for a discussion of Cottrell (1988), one possible connectionist model.

to make a judgement (which might be influenced by dynamic factors), and then selects the attachment favoured by the "winner" of this "race". Since all these processors are working at the same time, but independently, they may overlap without ill effect, making the system relatively robust with respect to which principles one chooses to include. In most cases all principles described earlier will be represented, but at times some may be so slow getting activated or so weakly supported that they do not make much difference except in sentences where all other preferences are relatively weak. Note that "races" will be confined to a particular set of attachment possibilities at a particular time and not allowed to extend through the rest of the sentence.

Encoding of Preferences

Given that preferences are to be encoded as timing functions, it remains to be seen how these functions are determined. Currently there appears to be no evidence as to the absolute time a particular hypothesis should take, although there is some empirical knowledge of the preferred interpretation of sentences in which preferences conflict. It is assumed that for present purposes it is sufficient to use intuitions about the relative amount of time they should take, tempered by knowledge of preferred structures. In addition, priming, either by semantics (related meanings, existence of referents) or syntax (parallel structures) may cause otherwise unpreferred structures to be preferred (Frazier et al., 1984; Hirst, 1987). In general, lexical hypotheses will be faster than syntactic hypotheses, which, in turn, will be faster than semantic or pragmatic hypotheses.

Lexical Preferences Lexical preferences, such as verb-frame preferences, determine the strength of expectations about complements that may follow. Thus, the stronger a lexical preference for an expected argument, the more quickly we are to assume that a given word begins a structure of the type we are expecting. Somewhere along the strength continuum for lexical preferences, these preferences become sufficiently weak, allowing other preferences to predominate. This point must be determined experimentally, which is beyond the scope of this thesis, so we will consider only clear-cut cases. The normative results of Clifton et al. (1984) and Connine et al. (1984) may be helpful in determining these cases.

Syntactic Preferences Minimal Attachment (MA), Right Association (RA), and Revision as Last Resort (RALR) are all preferences with respect to the building of syntactic structures. In order to encode these preferences we must insure that (in the absence of other preferences) non-minimal attachments, non-right associations, and attachments requiring revision take longer for structural hypothesizers to justify than structures that satisfy these preferences.

Recall that following the most general statement of MA means that among all possible attachments the one that is easiest to find and to build is the one that is favoured. Thus, to find the minimal attachment of a constituent W would be to find (in decreasing order of preference):

1. A place along the right edge of the current parse tree where a phrase beginning with W is allowed

2. A place along the right edge of the parse tree where a node of type W could be attached by first proposing some intermediate structure, *e.g.* a place in which a clause that begins a phrase that begins with W is allowed. For example we can attach the verb "lost" into the NP "the girl" if we first propose that the NP has a relative clause as a modifier.

Following RA means that given any ambiguity within either of the cases above, an attachment that links a word to adjacent words is favoured over an attachment that links a word to words that are further away. In other words, there is some cost associated with proposing a more distant connection, perhaps because accessing more distant words takes longer.

Following RALR means that attachments that require changing the current syntactic structure, rather than just attaching new structure directly to existing structure, are less favoured than attachments that do not require such changes. Moreover, it is proposed that certain revisions are more difficult, and hence more time-consuming, than others. For example, a revision that entails simply adding a recursive level of structure between two existing nodes will be easier than a revision that requires moving a piece of existing structure, and both of these revisions will be much easier than revisions that require multiple operations such as adding several levels of structure, moving a piece of existing structure, and inserting a trace. For example in sentence **(42)**, to combine the verb "fell" and the clause "the horse raced past the barn" after "raced past the barn" has been minimally attached as an active VP requires inserting a recursive $\overline{\mathrm{N}}$, adding a level of structure to represent the clause, inserting a trace to represent the subject of the clause, and moving the VP into this clause.

(42) The horse raced past the barn fell.

This revision may be so difficult that without other evidence (such as from some semantic expectation) the HSPM may just fail to recognize it, which would explain why most people garden-path on sentence **(42)**.

At this point a word must be said about dependence on grammar. In our discussion of MA, attachment classes were defined rather broadly to decrease the system's dependence on any subtleties of the grammar that have not been psychologically verified. We left minor differences in minimality to be resolved by other structural preferences such as verb-frame preferences and Right Association. Strictly speaking, incorporating RALR introduces some dependence on ad hoc features of the grammar because the grammatical theory allows that for some attachments, some grammars may construct a direct attachment whereas others may require a recursive level of structure. The solution to this problem is to make adding a recursive level of structure no more costly, *i.e.*, no slower, than a direct attachment, unless there is independent evidence to the contrary. For example, what the MA data show us is that it is generally less preferable (*i.e.*, harder or slower) to attach a structure as a modifier than as a complement. (To model this, a grammar might always attach modifiers using an extra, recursive level of structure.) Thus, the parser's implementation of RALR captures the major generalization that movement and significant restructuring is possible, but that they take so long that they will not be considered unless there is no simpler possibility, and allows, but does not require, one to make more subtle distinctions.

Semantic Preferences To simplify discussion, "semantic" preferences will be taken to be all those preferences that are neither lexical nor syntactic, *e.g.* preferences that occur because a certain attachment creates a description of an entity already in the knowledge base, or a description of an entity similar to one already in the knowledge base. Semantic processors thus attempt to match fragments of structure with the system's knowledge about the world, including previously processed structures.

At the beginning of this section I suggested that semantic preferences would be computed more slowly than lexical or syntactic preferences. This view is consistent with the results of Ferreira (1986) and Ferreira and Clifton (1986), who argued that semantics can only trigger or guide revisions— although since semantic preferences are determined by separate modules, the model could just as easily represent the converse approach: that semantics can be very fast and be used to guide initial attachment decisions (as argued by Crain and Steedman (1985)).

After selecting the preferred attachment, there are several options with respect to competing results. They could be saved for a limited amount of time (*i.e.,* until they are "forgotten" because of inattention), or they could be discarded in accordance with the limits on working memory. In either case this means that multiple interpretations are not pursued (see section 1.3.1) and that justifications are not necessarily tried in some fixed serial order (see Schubert (1986) for criticism of serial strategies).

Priming In the current model, priming may preactivate certain structures making modules that represent preferences for these structures run a little faster. In other words priming may alter some of the arguments of the timing function computed by an affected hypothesizer. At the lexical level, the lexical disambiguation process always checks for preactivation of a possible meaning or of a meaning closely related to a possible meaning, before seeing if other factors, such as selectional constraints, will help limit the number of choices. If a meaning has been sufficiently primed, then that meaning is chosen. (This strategy is compatible with the model's general view that the first possibility "recognized" is preferred, since, by assumption, "preactivation" occurs before lexical access, including the retrieval of selectional information.)

3.2.3 Revision

Since ambiguities are resolved on the spot using somewhat crude heuristics, the parser also requires a mechanism to handle failure. One approach to failure has been to "backtrack", which normally means going back to some selected point in the sentence (often the last choice point) and reparsing the sentence from that point forward, ignoring information about the state of the parse at which the difficulty was recognized. This forgetful form of backtracking is unappealing from both a psychological and a computational standpoint because it wastes the effort already invested in the current parse and ignores the parser's past experience with similar failures. Certainly some errors are very common and cause the human parser no difficulty whatsoever. For example, neither (43) nor (44) is difficult (Marcus, 1980).

(43) Have the students take the test.

(44) Have the students taken the test?

Marcus (1980) argues that they are both easy because the parser delays making a decision about their structure, but we have already argued that this is not plausible (see section 1.3.1). Instead, in these cases the sentence processor probably jumps to some standard conclusions and then uses some standard set of revision rules to correct itself later. This view is supported by Frazier and Rayner (1982) who found that when people discover that they have made an error in parsing, they immediately regress back to the point of ambiguity that the current input resolves. Some revisions will be more difficult than others (See section 3.2.2 for a discussion.)

Revision rules might be like the following:

> If there is a phrase XP with daughter nodes $X_i \ldots X_j$ along the right edge of the parse tree, but it was also possible to have attached $X_i \ldots X_j$ into the parse tree by supposing a phrase YP that allows $X_i \ldots X_j$ and the new node, then replace XP by YP and attach $X_i \ldots X_j$ and the new node.

3.2.4 Parallel Processing

Proceeding in parallel with syntactic processing are many other sentence processing subtasks (see section 1.3.3 and 3.4), but this thesis will focus on just two of them: lexical disambiguation and semantic processing.

Lexical Disambiguation

Sentence processing is complicated by the fact that many of the words we use have multiple meanings. This problem has been widely noted, and addressed in other research (see Hirst (1987) and references therein). Lexical disambiguation cannot occur prior to syntactic processing because words that are categorially ambiguous (*e.g.* "sink" can be a noun or a verb) require syntactic context to resolve the ambiguity. My sentence processor incorporates Hirst's "online" approach to lexical ambiguity resolution called "Polaroid Words" (Hirst and Charniak, 1982; Hirst, 1984; Hirst, 1987; Hirst, 1988).

As each word is read, a special object is created for it representing all possible senses of that word, and a Polaroid Word process (PW) activated for it. The PW is responsible for the disambiguation of the word. A PW uses information such as activation by semantic context and selectional contraints among neighboring words to try to resolve any ambiguity. Moreover, the activation of one PW will trigger the activation of its neighbors, so they, too, may use knowledge of the changed semantic context and newly introduced constraints to help them with their disambiguation task. A PW will be repeatedly reactivated until neither it nor any of its neighbors can eliminate any more meanings, and then "rest" until the creation of a new PW triggers new activity. Syntactic processing may interact with lexical disambiguation by causing a PW to eliminate all the senses that correspond to lexical categories incompatible with the current syntactic context. One could also allow lexical disambiguation to help syntactic processing by permitting rejection of categorial possibilities for semantic reasons.

Related to the problem of lexical ambiguity is the ambiguity associated with verbs and their θ-roles, (roughly equivalent in this context to "thematic roles", "case roles" or "case slots"; see section 4.2.1) (Carlson and Tanenhaus, 1987). For example, associated with the "split into pieces" meaning of the verb "break", there are the following "θ-grids", *i.e.,* ordered sets of thematic roles:

(a)	({Agent}, {Theme})	*e.g.,* John broke the window.
(b)	({Instrument}, {Theme})	*e.g.,* The rock broke the window.
(c)	({Theme})	*e.g.,* The window broke.

PWs have been used to disambiguate θ-roles (Hirst and Charniak, 1982; Hirst, 1987) by having the parser insert "pseudo-prepositions" before each of determiner phrases that the verb assigns "case" to (see section 4.2.1), and letting the PWs for these tokens decide which θ-role they should have as their meaning (*e.g.* Agent, Instrument, or Theme). In order to remain compatible with GB, however, my system disambiguates θ-grids within the PW for the verb itself. Thus, the PW associated with a particular verb includes a list of possible θ-grids. When a DP is attached to a position in the parse tree that receives a θ-role from the verb, it receives an index indicating the θ-role's location in the verb's θ-grid (Di Scuillo and Williams, 1987). These indices can later be used to retrieve the name of the slot the DP fills in the frame representation of the verb (as discussed in the next section). [4] Then, a verb sense checking its compatibility with a particular noun requires that at least one of its θ-grids be long enough so that the noun's index does not go out of range, and that the noun be compatible with the selectional restrictions of the θ-role at the indexed location in the θ-grid. Any θ-grid that fails to meet either of these two conditions can be eliminated from consideration. A verb sense checking its compatibility with a preposition might check that the preposition does not duplicate a previously assigned θ-role, and perhaps eliminate incompatible grids.

Semantic Processing

As Polaroid Words are disambiguating words, and the parser is building syntactic structures, the system is also computing semantic representations of the constituents that have been received. The semantic representation scheme used by the system was first developed for use by the semantic interpreter Absity (Hirst, 1983; Hirst, 1987) (also see section 2.3). The representations are immediate descriptions of semantic objects, *i.e.,* frames, or frame-related objects, in the knowledge base (see section 4.4). The representations of words and phrases are determined by their "meaning" and by their semantic type, where the "meaning" is just an unambiguous name for the concept being represented by a possibly otherwise ambiguous word, in a given context, and the semantic type is the name of the frame-related object the word or phrase corresponds to (see figure 3.1). As in Absity, for every syntactic attachment operation there is a corresponding semantic operation, and when an attachment is performed, the corresponding syntactic and semantic operations are executed in tandem. The semantic types of objects determine how their semantic representations are combined.

The exact correspondence between semantic and syntactic types is dependant on the actual grammar assumed. For example, the syntactic structures and categories used

---

[4] Using indices also allows us to extend PWs to handle passivisation, which can be treated as an operation either on θ-grids (Cottrell, 1988) or on the assignment of indices (Jackendoff, 1987).

Syntactic Type	**Semantic Type**	**Example Semantic Representation**
Determiner	Frame determiner	`(the $x)`
Noun	Frame descriptor	`(apple $x)`
Adjective	Slot–filler pair	`(color $x red)`
Preposition	Slot name	`location`
Verb	Frame descriptor	`(put $x)`
INFL	F-infl	`<PAST>`
Noun Phrase	Frame descriptor	`(apple $x (color $x red))`
Determiner Phrase	Frame statement	`(the $x (apple $x (color $x red)))`
Prepositional Phrase	Slot–filler pair	`(location $x (the $y (table $y)))`
Verb Phrase	Frame descriptor	`(put $x (theme $x (the $y (apple $y))) (location $x (the $z (table $z)))`
INFL Phrase	Frame-statement/ frame-determiner	`(<PAST> (read $x (agent $x John) ...))`

Figure 3.1: Some Example Semantic Types and Representations [5]

in GB theory are not the same as the ones used in the grammar assumed by Absity. Additional semantic types must be added to account for INFL and its projections because the category INFL subsumes both elements that were slots (such as "to") and elements that were slot-filler pairs (such as the tense marker). The new types must account for the fact that INFL is used in different ways depending on the context, while preserving the strong typing of syntactic categories. (See figure 3.1.) Thus IP either can be used to build a slot-filler pair (with INFL acting as the slot) when the IP is an infinitival clause (such as "to pass the time") being added to a verb phrase or it can be used as to build a frame-descriptor (with the INFL moved inside filling the TENSE slot of the verb) when it is being added to a complementizer phrase. Appendix A contains a complete list of the syntactic types used in the grammar the system uses and their corresponding semantic types.

3.3 Incorporating Expectations

One of the strengths of the original Sausage Machine (SM) model is its use of syntactic expectations. (See section 2.4.2 for a discussion.) My parser also makes predictions about what is to follow, as well as making a few "backward predictions" about what null (syntactically subatomic) elements must have been present. In particular, it incorporates Barbara Brunson's proposal (personal communcation) that, in addition to the conventional predictions that can be made about the expected arguments of a word (*e.g.* a preposition expects a determiner phrase), there are also predictions we can make from Case Filter constraints (*e.g.* a determiner phrase expects INFL[6] or a possessive marker).

---

[5] Symbols beginning with "$" are variables.

[6] INFL is the head of the clause, according to current accounts of Government-Binding theory (Cowper, 1987). It is realized either by the word "to" indicating that the clause is infinitival, or by the affix on

In addition, there are backward predictions that can be made because syntactic expectations and the morphology act together to license the presence of a syntactically subatomic plural determiner or INFL (*i.e.*, one present only as an affix), which the parser may then insert into the appropriate location in the structure. Making these backward predictions is necessary because, for reasons of simplicity, the implementation does not incorporate morphology as a separate process running in parallel. (In the next section, we will mention some other ramifications of this simplification. Note that it is *not* sufficient to assume that there is a front-end morphological analyser that recognizes INFL in the verbal morphology and "undoes" the effects of Affix-hopping as in Abney and Cole (1985), because the existence of INFL is dependent upon the syntactic context; there must be a prior expectation for INFL.)

3.4 What's Missing

In order to tackle the problems set forth in this thesis, several important aspects of sentence processing were excluded or covered in only a superficial way. Among the most important aspects not given a complete treatment is morphological analysis. Presumably, morphological analysis goes on in parallel with other processes, with richness comparable to syntactic processing, but the results of this analysis are atomic with respect to syntactic processing (see Di Scuillo and Williams (1987)). Thus, we can, for the most part, substitute a rich and perhaps redundant lexicon for a full-scale morphological analyser. So for example, word compounds like "jet black", which acts as an adjective, will be listed in the lexicon as "jet-black". Another approach would be to index compounds in the lexicon according to their initial word and then employ a simple matching process (Wehrli, 1984). Our approach has the advantage that it safely allows us to ignore much of the problem of noun-verb ambiguity (see Milne (1982) and Arens et al. (1987)), which, while important, is orthogonal to the issues being addressed here.

Idiomatic expressions, which are syntactic phrases listed in the lexicon, have also been placed outside the scope of this thesis. Idioms differ from word compounds in that idioms have a syntactic form or phrase structure, but their meaning is not determined compositionally. To handle them, one might add a phrasal lexicon and compare completed phrases with listed phrases. If a match was discovered then one could just replace the semantic representation computed compositionally with the representation found in the lexicon.

I have also not addressed many important syntactic issues such as gap filling and resolution of anaphors. This means that I have not been concerned with incorporating binding theory.

At the interpretation end, the processor produces a simple semantic representation, but I will not say much about how that representation is to be used by the knowledge base. So for example, I will not say anything about how information from the interpretation of tense and aspect is used or much about what inferences a sentence should trigger. Arguably this is where most (if not all) of the "meaning" of sentences lies, but again, these problems are independent of those being addressed by this thesis.

---

the first verb when the clause is tensed.

Chapter 4

The Implementation

4.1 Syntactic Processing

Like the original Sausage Machine model, my parser is syntax-driven. In other words, processing proceeds according to the expectations of syntax. Semantic and thematic processing proceed in parallel with syntactic processing; normally the operations they execute correspond to the syntactic operations being done, although sometimes their results may influence syntactic decisions.

The general algorithm for the first stage of the syntactic processor is as follows:

1. Get the next word (*i.e.,* the next syntactically-atomic unit) and activate a lexical disambiguation process (Polaroid Word) for it.
2. Incorporate the word into the current package of phrase structure. If there are words already in the current package, then combine the new word with preceding constituents according to existing expectations, selectional requirements of the new word, known structural constraints (*e.g.* the direction of headedness in English), structural preferences, semantic information, or knowledge of possible revisions.
3. Check for elements licensed by morphology or predicted by case theory and add them to the current package, if necessary.
4. Check memory status. If the first stage is reaching capacity and there is a good place to end the package, close it off and pass it to the second stage.

The general algorithm for the second stage of the syntactic processor is as follows:

1. As each package is created by the first-stage processor, attach the package to the current structure, if there is any, using existing expectations, selectional restrictions, structural constraints and preferences, semantic information, and knowledge of possible revisions.

4.1.1 The Attachment Process

The attachment process is governed by the generalized statement of Minimal Attachment, that is, always perform the first acceptable attachment that is identified. For each attachment there is a set of objects, henceforth called "hypothesizers", each of whose job it is to identify a set of possible attachments based on a particular justification or preference (for example lexical, syntactic, or semantic). Thus, when the parser wishes to attach two objects, it consults the hypothesizers, and evaluates their most favoured attachment.

The general algorithm for selecting an attachment is as follows:

1. Get all the hypothesizers that apply to the current input (which may be a word or a larger constituent). For example, if the input is a noun, then retrieve all the hypothesizers that know how to attach nouns.
2. Activate all the selected hypothesizers and for each one get the list of possible attachments it has identified. These possibilities are each described by means of a vote consisting of a time value, a goodness rating, and a description of the attachment. (In the current implementation, these descriptions are represented as an expression that, when evaluated, will construct the described attachment.)
3. Find the vote with minimum time value whose goodness rating is above the threshold of acceptability. Declare this vote the "winner".
4. Execute the "winning" attachment.

The model leaves open the question of what to do with the "losing" attachments. One could just ignore slower competitors, permitting the parser to later use their results to guide a revision process if necessary. In a parallel setting where efficiency was an issue, the first hypothesizer finished might kill off (or suspend) all slower competitors to prevent wasted effort. Similarly, in a serial implementation in which one wanted to prevent wasted effort, one could give each hypothesizer a time limit and constrain it to not even attempt any processing that would require additional time. This time limit could be increased gradually until an acceptable attachment is received. (Such a technique is often called "time slicing".)

The Attachment Hypothesizers

The hypothesizers themselves are a combination of the general and the specific across parts of speech. In this way we can handle certain special cases (such as subjects that consist only of a categorially ambiguous plural noun) and exploit any generalities (such as attaching words according to selectional information). Since hypothesizers run in parallel and compute their time value internally, it is simply an implementation detail as to how general or specific hypothesizers are, permitting us to make the choice that is easiest to program or most efficient to run.

The current implementation includes the following hypothesizers (in which X and Y are two elements such that Y immediately follows X in the buffer of either stage of the parser (and hence in the sentence)):

1. A hypothesizer that checks for explicit expectations that would link two items: For example X may expect Y as an argument or Y may expect X as a specifier, according to the grammar. Information about such syntactic expectations is contained in the lexicon. These are minimal attachments and so are very rapid.

2. A hypothesizer that recognizes that when a sentence begins with a word that can be either a plural noun or a verb, the subsequent input of a verb disambiguates the first word and creates a chain of expectations that will link the two words; specifically, the plural noun morphology licenses a null determiner, the verbal morphology licenses INFL, and INFL selects a determiner phrase specifier (DP) and a verb phrase argument (VP). All this information is encoded into this special-purpose hypothesizer and the corresponding function that implements the attachment. This is less rapid than the previous hypothesizer.

3. A hypothesizer for attaching post-modifiers: An item X allows item Y as a post-modifier if, firstly, there is a phrase structure rule (see section 4.2.2) linking Y to a projection of X as a post-modifier (*e.g.* $\overline{X} \leftarrow \overline{X}\ Y$), and secondly, X modified by Y describes an entity that is known to either exist or possibly exist (Hirst, 1987). These attachments are presumed not to be expected arguments and hence are "nonminimal". They are also subject to RA and hence will require increasing time with increasing distance. Thus, unless there is priming (which reduces the time needed to check existence), they will also require more time than syntactically expected attachments.

4. A hypothesizer for attaching pre-modifiers: An item Y may be attached to item X if a projection of X expects an argument Z and there is a phrase structure rule linking Y to Z as a pre-modifier. This is less rapid than an expected attachment.

5. A hypothesizer that checks for thematic expectations that would link two items (*i.e.,* Y is an expected argument or modifier of X). Thematic expectations are determined by a separate "argument processor" (see section 4.3) that reports the degree to which the argument is expected. These attachments are the result of verb-frame preferences and thus may be more strongly favoured than other attachments when there is no priming. The time required by the hypothesizer reflects the fact that more highly expected items are somehow easier to access, or more readily assumed to exist, than less expected items. For highly expected items, these attachments will be equally as fast as for syntactic expectations.

6. A hypothesizer that detects start of reduced relative clause. This is a nonminimal attachment involving some revision of the current structure and hence will take quite some time unless there is some sort of priming.

7. A hypotheszer that detects missing morphology: Syntactically subatomic elements such as null determiners and INFL (the head of the clause) are realized (or *licensed*) by morphological affixes on nouns and verbs (plural marking and tense marking, respectively). (see section 3.3) This type of attachment requires recognition of the possible licensing, creation of a token to represent the subatomic element, and then attachment of both the token and the item the system was trying to attach in the first place. This hypothesizer thus requires a great deal of time, making it favoured only when there are no other possibilities detected.

The time value associated with each vote is intended to reflect the relative amount of time it takes to identify the attachment it describes. For some justifications or preferences the difficulty of identification process is related to the difficulty of actually constructing the proposed attachment, so the time value should be proportional to the amount of time necessary for construction. Time values may be reduced by priming, for example to reflect faster access of information used to justify attachments such as existing referents. Time values may also be adjusted to simulate (or adapt to) individual differences in the use of different types of heuristics; for example, Aaronson (1986) and Bever et al. (1987) describe differences in their subjects' reliance on structural representations compared to their reliance on lexical or conceptual knowledge. Adjustments could also be made to compensate for people's apparent tendency to change their processing strategies (namely, increasing their preference for RA relative to their preference for MA) when they communicate with computers instead of other people (Guindon, Shuldberg and Conner, 1987).

4.1.2 The "Shunting" Process

Once the size of a package approaches the capacity of the first stage, the parser starts looking for a good place to end the package and shunt it off to the second stage. Good places to end a package are just before the start of a new constituent, usually signalled by the presence of a grammatical function word (*e.g.* a determiner or preposition) (Fodor and Frazier, 1980), just before a pause, usually signalled by a punctuation mark (Macdonald, 1979), or just before an item that cannot be integrated into the current package easily (*e.g.* a time adverb that does not match the tense of the verb phrase just constructed), although this last case would normally be an instance of the second possibility as well. Bad places to end a package are places in which there is an outstanding syntactic expectation (such as between a preposition and its determiner phrase complement), or just before the last word in a sentence.

Currently, the parser just checks whether there are at least two words left and whether the next "word" is a punctuation mark, a determiner, a complement, a preposition, a conjunction, a verb, or INFL, handling the problem of outstanding expectations by making the shunting process only "semi-discrete". Recall that shunting a package to the second stage causes the first stage to forget the contents of that package. There are, however, relationships between words and their arguments, such as case assignment, that are very important to retain. The parser handles this by representing open constituents in a stage as separate units in the buffer of that stage. Thus, when shunting occurs, only the highest-level constituent is shunted, preserving its attachments to lower-level open constituents. This means that for a time a structure may extend across the stage boundary.

4.2 About the Grammar

The grammar of the implementation is derived from Government-Binding theory. The parser builds structures according to $\overline{\text{X}}$-theory and its knowledge of the direction of headedness in English. $\overline{\text{X}}$-theory (parameterized for English) predicts that all phrases will have the structure shown in figure 4.1 (including some optional levels), with parameterization simply fixing the order of the branches at each level. The system considers nouns, verbs, adjectives, adverbs, prepositions, determiners, and INFL to all be heads,

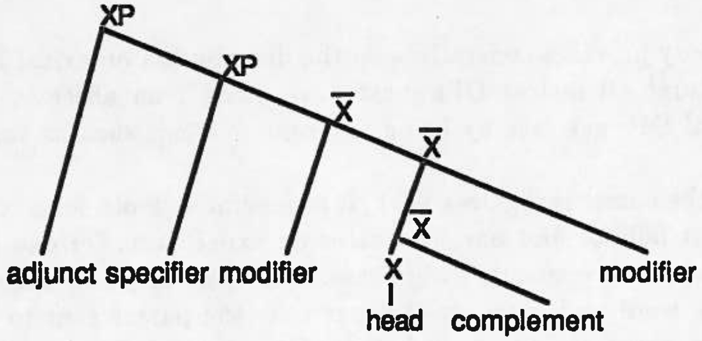

Figure 4.1: $\overline{X}$ Structure

and assumes the maximal bar level for each head to be two.

In addition, a word may be restricted to occuring only in certain contexts; in particular the word may require (or prohibit) a specifier or some complements, and may also require them to have certain features if they are present. (Note that adjuncts and modifiers are always optional.) These restrictions are contained in the word's lexical entry, an element of the lexicon. A word has much less control over the words not in its immediate context (*i.e.*, not its sisters), for example, its modifiers. Finding a good theoretical account of what can modify what from what direction is still an open question in linguistics, and thus the parser relies on a limited set of phrase structure rules to describe this information.

4.2.1 The Lexicon

Following Brunson (1988), the lexicon is divided into two parts, the word lexicon and the category lexicon. The word lexicon associates each word with its "meanings" (see section 3.2.4) and corresponding syntactic categories. The category lexicon associates each category with its semantic type (see section 3.2.4), as well as default selectional information for members of that category. The lexicon is implemented in MRS, a Prolog-like knowledge representation language (Russell, 1985).

As might be expected, the lexicon is very important to the semantic interpretation process, because semantic representations are constructed according to meaning and semantic type. In addition, since syntactic structures are constructed by projecting lexical heads and attaching the projections together on the basis of selectional information stored in the lexicon, the lexicon also plays a central role in syntactic analysis. In terms of Goverment-Binding theory, the lexicon includes information that subsumes much of Case theory and θ-theory.

Case Theory

In GB, Case theory provides constraints on the distribution of lexical DPs, by requiring that at S-structure[1] all lexical DPs must have "case", an abstract property of constituents. Lexical DPs get case by being adjacent to items such as verbs, prepositions, and INFL.[2]

Thus, when the parser recognizes a DP, it immediately looks for a corresponding case assigner, and if it fails to find one, it creates an expectation for one (Brunson, 1988). Information about what elements assign case, and what they can assign it to, has been encoded into the word and category lexicons that the parser uses to build structures. In particular, the category lexicon includes information about the specifiers and complements selected by lexical categories, and the parser sees to it that these constituents are adjacent to the phrasal head. A categorial lexical entry includes a phrase-grid of the form: (*specifier category complement*). The category lexicon includes default phrase-grids for the categories determiner, complementizer, preposition, and INFL as shown in figure 4.2. Now, to account for the fact that some verbs have arguments that are optional or that can be one of a set of syntactic types, there are also mechanisms for indicating optionality and "enumerated" types; namely, optional arguments are listed in parentheses and enumerated types are listed in a parenthesized list beginning with the token "OR". For example, the lexical entry for the verb "break", which optionally selects a DP complement, might look as shown in figure 4.3. (I will explain the meaning of "th-grids" and "th-roles" in the next section; "rating" is just a measure of the relative frequency of usage among the different meanings.) The elements of a phrase-grid are taken to be obligatorily adjacent in the sentence (unless marked as optional), and in the order given.

determiner	`(DP det NP)`
complementizer	`(nil comp IP)`
preposition	`(nil prep DP)`
INFL	`(DP infl VP)`

Figure 4.2: Default Phrase-Grids

Nouns, verbs, and adjectives all have their selectional properties (which may be described by a phrase-grid similar to general category entries) stored with individual words. The word lexicon includes modifications to the default category entries (for example, the fact that "the" does not select a specifier, or that "cat" is singular), syntactic features such as number and case, as well as semantic information. For example, an entry for the noun "bank" might look as shown in figure 4.4.

---

[1] S-Structure is the surface syntactic structure, annotated with traces and empty constituents.

[2] The case assigners in GB are the non-lexical category INFL, and the lexical categories that have the feature [−N].

```
(break verb  ((form infinitive) (case))
              (meaning:  BREAK
               rating:   20
               cases:    (nil break (DP))
               th-grids: ((Agent Theme) (Instrument Theme) (Theme))
               th-roles:
                         ((role:    Agent
                           sr-reqs: ANIM)
                          (role:    Theme
                           sr-reqs: PHYSOBJ)
                          (role:    Instrument
                           sr-reqs: PHYSOBJ))))
```

Figure 4.3: Sample Lexical Entry for a Verb

```
(bank noun  ((number singular))
            (meaning: BANK-BUILDING
             rating:  20
             isa:     BUILDING)
            (meaning: STEEP-SLOPE
             rating:  10
             isa:     NATURAL-LOCATION)
            (meaning: FINANCIAL-INSTITUTION
             rating:  20
             isa:     ORGANIZATION)
            (meaning: RIVER-EDGE
             rating:  20
             isa:     NATURAL-LOCATION))
```

Figure 4.4: Sample Lexical Entry for a Noun

θ-Theory

θ-theory provides semantic information about the roles that various arguments bear to the word that they complement (see Sells (1985), Brunson (1986), and references therein). In particular it is intended that there be a one-to-one mapping between arguments and θ-roles as expressed in the θ-Criterion:

> Each argument bears one and only one θ-role and each θ-role is assigned to one and only one argument (Chomsky, 1981: 36)

where arguments of a lexical head may be internal or external to the corresponding phrase; for example, the verb "buy" requires two θ-roles: {Agent} and {Theme}.[3] Thus, among GB theorists it is generally agreed that associated with each verb in the lexicon there is an information template (called the argument structure or θ-grid) listing the thematic roles of the verb, perhaps along with some information about the syntactic structures that may bear those roles. Similarly, it is agreed that NPs inside PPs receive their θ-role via the preposition.

θ-role information has been encoded into the word lexicon of the sentence processing system. For prepositions, the θ-role of its NP argument will be identified with the set of possible "meanings" of the preposition itself. (In the interest of simplicity, we currently ignore the semantic differences between different prepositions of the same type, such as "in" and "on".) For verbs, we include a list of θ-grids, which are lists of θ-roles, and a list of selectional restrictions for each role. Thus, for example, the lexical entry for the verb "put" includes a structure like the one shown in figure 4.5.

```
(th-grids: ((Agent Theme Location))
(th-roles:
            ((role:    Agent
              sr-reqs: ANIM)
             (role:    Theme
              sr-reqs: PHYSOBJ)
             (role:    Location
              sr-reqs: (or PHYSOBJ VOID)))
```

Figure 4.5: Sample Thematic Information found in the Lexicon

4.2.2 Phrase Structure Rules

Although it is relatively straightforward to encode information about the arguments of words in the lexicon, encoding information about possible modifiers is not so easy. In particular, one must devise a scheme for determining which types of categories can modify which types of phrases, and on which side of the lexical head they can occur. For English, one can generally state that modifiers follow (*i.e.*, occur on the right-hand side of) the head, but one-word adjectives and adverbs seem to be an exception. Rather than take a stand on this particular linguistic issue, this research incorporates a limited set of phrase structure rules to describe modification, including the following:

---

[3] θ-roles, although often identified with a single thematic relation as in the above example, are actually non-empty *sets* of thematic relations as demonstrated by the following sentence:

(i) Tom gave the book to Diane.

where the subject NP "Tom" is both the agent and the source. This observation is credited to Jackendoff (1972).

$\overline{N} \rightarrow \overline{N}$ PP
$\overline{N} \rightarrow \overline{N}$ AdjP
$\overline{V} \rightarrow \overline{V}$ PP
$\overline{I} \rightarrow \overline{I}$ Adv
$\overline{N} \rightarrow$ Adj $\overline{N}$
$\overline{I} \rightarrow$ Adv $\overline{I}$
IP $\rightarrow$ PP IP

Thus, when an unexpected word arrives, one way we attempt to attach it is by finding a sequence of rules that links the current expectation or partial phrase marker to the incoming word. We judge this attachment by comparing the result with objects we have seen in the past. The existence of an object matching or nearly matching the resulting description is taken as evidence in support of trying this attachment again, accounting for semantic and structural priming.

4.3 Argument Processing

In addition to the obligatory arguments of words, some words (most notably verbs) have optional arguments or adjuncts. Distinguishing optional arguments from modifiers can sometimes be difficult, however. Roughly, optional arguments change the meaning of the verb in some way, whereas modifiers are more circumstantial. For example, many verbs allow a Beneficiary, commonly considered an optional argument. "Gave" is one such verb, as shown by (45), in which "Evan" is the Agent (and Source), "the book" is the Theme, and "Mara" is the Beneficiary (and Goal).

(45) Evan gave the book to Mara.

Similarly, most verbs permit a Temporal, such as "today", usually considered a modifier, as in (46).

(46) Murray sent more electronic mail today.

Nevertheless, there do not seem to be class distinctions between arguments and modifiers (Vater, 1978), and deciding whether something is an optional argument or a modifier can be problematic. (See Somers (1987) for a discussion.)

A related difficulty is deciding which, if any, of a verb's arguments belong in its lexical entry. It seems sensible that all obligatory arguments be in the lexical entry, because they are small in number (at most three, including the subject), and definitely expected. It is also a good idea to include the arguments to which the verb assigns case (see section 4.2.1) because they are also the arguments that must get their θ-roles directly from the word they complement. The remaining arguments are "oblique" phrases such as AdvPs and PPs. One could attempt to include all of these arguments, but this approach introduces at least two problems: firstly, it may easily result in a lot of lengthy lexical entries and no obvious stopping point, and secondly, since oblique arguments can occur in almost any order, and in any combination, one would have to either include an exponential number of ordered θ-grids, or use unordered grids and provide some additional mechanism for associating arguments with θ-roles. To avoid these problems, the system presented here exploits the apparent fact that there exist certain classes of verbs such that all the verbs in the class permit the same optional arguments, and includes a mechanism for adding these optional arguments to verb phrases as they occur. For example, verbs of movement may all expect the following optional arguments (Fink, 1978):

(a) Path	*e.g.*, into the sea
(b) Environment	*e.g.*, on land
(c) Means of Conveyance	*e.g.*, by car
(d) Modes of Action in Locomotion	*e.g.*, momentary
(e) Velocity	*e.g.*, quickly
(f) Manner of Locomotion	*e.g.*, sneakily

Given a verb phrase and a potential argument or modifier, a redundancy rule system such as the "argument processor" used by my system can check for arguments already satisfied by constituents in the sentence or by the verb itself (*e.g.* "swim" lexicalizes the Environment "in water"), and indicate whether the verb currently expects the potential argument or modifier. In addition, certain arguments and modifiers may be more highly expected than others, and the argument processor can indicate this also.

These differences in expectedness we have referred to as verb-frame preferences (see section 1.3.2). Information about these preferences will be encapsulated in the argument processor. The sentence processor will ask the argument processor whether a verb (as a semantic representation) allows a certain slot, and the argument processor will respond with a numerical value indicating "yes" or "no", which is usable as an argument to the sentence processor's timing function. Thus, a strong "yes" response will have a relatively low value and a strong "no" response will have a relatively high value. Intuitively this value corresponds to the closeness of the association, where closeness may be determined not only by number of links in some neural network, but in the strength of the connection, since weak connections are presumably activated more slowly than strong ones. As an approximation, the argument processor will determine its response value according to a fixed rating scale.

4.4 The Knowledge Base

The system's knowledge base (KB) encodes information about the objects the system has encountered and the way objects are related to each other in the real world. The KB includes a hierarchical classification of objects that the system knows about (*e.g.*, that a bank is a building and a building is a location constructed by humans), along with instances of these objects, and inferences it has made about them.

The underlying knowledge representation language for the system's current KB is MRS[4] a Prolog-like language with meta-level reasoning capabilities, developed at Stanford (Russell, 1985) and implemented in Lisp. Additionally, the system uses a set of frame representation constructs implemented on top of MRS to approximate the syntax of the Procedural Semantics Network (PSN) language (Becker and Greiner, 1987; Levesque and Mylopoulos, 1979).

---

[4] The name "MRS" has had several different meanings, including Meta-level Representation System.

The contents of the KB can be described in terms of decreasing levels of generality. At the most general level of the KB is a taxonomy of basic concepts, encoded in PSN, as in the following:

```
(assert '(class-put class thing-class nil (class) nil))
(assert '(class-put thing-class thing nil nil nil))
(assert '(class-put thing-class LOCATION nil (thing) nil))
(assert '(class-put thing-class ARTIFICIAL-LOCATION nil (LOCATION) nil))
(assert '(class-put thing-class BUILDING nil (ARTIFICIAL-LOCATION) nil))
```

(This set of statements defines a special class of objects called "thing-class" hierarchically structured such that all BUILDINGs are ARTIFICIAL-LOCATIONs, all ARTIFICIAL-LOCATIONs are LOCATIONs, and all LOCATIONs are things, where "thing" is the most general or "greatest" element in the hierarchy. The first statement defines the meta-class "thing-class", a type, or "instance", of a class. The following four statements define classes which are instantiations of thing-class. Each definition of a new class includes the name of the meta-class being instantiated, the name of the new class, a list of the values for its slots, a list of its "is-a parents" in the hierarchy, and a list of slot definitions for its own instances.) Below this taxonomy of basic concepts is the lexicon (see section 4.2.1) which includes the "linguistic" concepts, *i.e.*, concepts that the system has words for. Some of these concepts may be a part of the taxononomic level also, but many others will be specializations of known concepts. Finally, at the most specific level are instances of linguistic concepts, *i.e.*, individual objects and events the system has learned about while processing sentences, encoded as PSN frames.

Recall that the semantic representations that the sentence processor creates are intended to be immediate descriptions of objects in the knowledge base. A complete description will thus consist of a semantic object that, when evaluated, returns an instance of an object in the KB. These descriptions we call "frame statements" and they consist of a frame retrieval procedure ("frame determiner"), and a description of a frame ("frame descriptor") compatible with the expectations of the retrieval procedure. As sentences are processed, any completed descriptions (which may be components of more complex descriptions) may be evaluated with respect to the KB. This information can then be used to assist later processing or to answer questions about earlier processing, *e.g.*, to decide whether a proposed structure describes an entity that either exists or might exist. (Existence may be determined by attempting to retrieve an object from the KB on the basis of a complete semantic description; similarly, plausibility may be determined either by attempting to retrieve an object from the KB or by matching a class description in the KB on the basis of an incomplete semantic description in which the unspecified parts have been left as variables.)

The KB can also include rules of inference for reasoning about what has been processed. For example, if one were to store a fact like "Mary bought an apple.", one would also like to infer the fact that Mary now owns an apple. This can be done in MRS by adding a forward-chaining rule such as the following:

```
(defrule '(if (and (BUY $x)
                   (Agent $x $y)
                   (Theme $x $z))
              (OWNS $y $z))
          '(direction forward))
```

which says that if the system ever creates an instance of a BUY frame with an Agent and a Theme, then it should add the fact that the Agent OWNS the Theme. More importantly, however, the system is able to describe the θ-roles of a particular verb in terms of a single thematic relation, and then let the KB infer the remaining relations to obtain the complete θ-role. For example, a rule like the following encodes the fact that the Agent of a GIVE frame is also the Source:

```
(defrule '(if (and (GIVE $x)
                   (Agent $x $y))
              (Source $x $y))
          '(direction forward))
```

Encoding θ-roles in the KB as part of the system's knowledge of the world allows us to use some simplified versions of semantic representations in the lexicon.

4.5 Some Examples

The goal of the system is to input English sentences and to output a corresponding syntactic and semantic representation for each sentence. The syntactic representations that the system builds are in accord with current versions of Government-Binding theory (Cowper, 1987; Abney, 1987; Brunson, personal communication). The semantic representations are in accord with Absity (Hirst, 1983; Hirst, 1987), discussed in section 3.2.4. Some examples follow.

Input Sentence:

```
Tom bought the book for Diane.
```

Syntactic Representation (See figure 4.6 for a diagram.):

```
(comp (bar 2)
    (comp (bar 1)
      (comp (bar 0) nullcomp)
      (infl (bar 2)
        (det (bar 2) tom)
        (infl (bar 1)
          (infl (bar 0) agr)
          (verb (bar 2)
            (verb (bar 1)
              (verb (bar 1)
                (verb (bar 0) bought)
                (det (bar 2)
                  (det (bar 1)
                    (det (bar 0) the)
                    (noun (bar 2)
                      (noun (bar 1)
                        (noun (bar 0) book))))))
              (prep (bar 2)
                (prep (bar 1)
                  (prep (bar 0) for)
                  (det (bar 2) diane)))))))))
```

Semantic Representation:

```
(a1 $v18 (buy $v18 (agent $v18 (the $v12 (hanim $v12 (name $v12 tom))))
                   (theme $v18 (the $v13 (book $v13)))
                   (benef $v18 (the $v16 (hanim $v16 (name $v16 diane))))
                   (tense $v18 <past>)))
```

Sample Derivation:

```
Input:  mark read the letter to chrysanne.

Reset knowledge base? (Y or N) Yes.


Word received:  mark

Attach:
        (det (bar 2) mark)

The buffer is empty, so we just add maximal projection to buffer.
Result:
  (det (bar 2) mark)
```

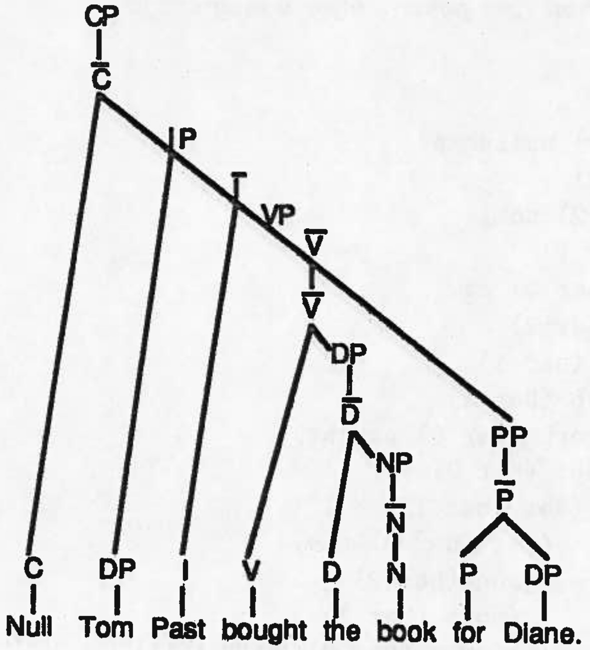

Figure 4.6: Syntactic Parse According to System's Grammar

```
Morph expectations:  None.
Case expectations:  Item needs case.

Result:
  (unset (bar 2)
    (det (bar 2) mark)
    (Expects:  (OR: ((det (bar 1) ((case))) (infl (bar 1) ((case))))))

Word received:  read

Attach:
        (unset (bar unset) read)

Call hypothesizers:   hypoth1 hypoth2 hypoth6 hypoth7

Votes returned:
  Vote number 1 by hypoth7 (detects missing morphology)
    Action:  (funcall (function reduced-rel-rule) @stack 0)
    Time:  6
    Goodness:  100
  Vote number 2 by hypoth7 (detects missing morphology)
```

```
    Action:  (funcall (function infl-rule) @stack 0)
    Time:  3
    Goodness:  100

Winning vote:  vote number 2

Morphology licenses INFL.

Attach:
        (infl (bar 0) agr
          (Specifier:  (det (bar 2)))
          (Expects:  (verb (bar 2))))

Call hypothesizers:   hypoth1 hypoth3 hypoth5

Votes returned:
  Vote number 1 by hypoth1 (detects expectations)
    Action:  (attach-new-word @stack 0 0)
    Time:  1
    Goodness:  100

Winning vote:  vote number 1

Attach:
        (verb (bar 0) read
          (Expects:  ((optionally) det (bar 2))
                     ((optionally) det (bar 2))))

Call hypothesizers:   hypoth1 hypoth2 hypoth6 hypoth7

Votes returned:
  Vote number 1 by hypoth1 (detects expectations)
    Action:  (attach-new-word @stack 0 0)
    Time:  1
    Goodness:  100

Winning vote:  vote number 1
Result:
  *** Node printed twice because it is still open. ***
  (verb (bar 2)
    (verb (bar 1)
      (verb (bar 0) read))
    (Expects:  ((optionally) det (bar 2) at verb (bar 1))
               ((optionally) det (bar 2) at verb (bar 1))))
```

```
  *** Node printed twice because it is still open. ***
  (infl (bar 1)
    (infl (bar 0) agr)
    (verb (bar 2)
      (verb (bar 1)
        (verb (bar 0) read))
      (Expects:  ((optionally) det (bar 2) at verb (bar 1))
                 ((optionally) det (bar 2) at verb (bar 1)))))

  (infl (bar 2)
    (det (bar 2) mark)
    (infl (bar 1)
      (infl (bar 0) agr)
      (verb (bar 2)
        (verb (bar 1)
          (verb (bar 0) read))
        (Expects:  ((optionally) det (bar 2) at verb (bar 1))
                   ((optionally) det (bar 2) at verb (bar 1))))))


Morph expectations:  None.
Case expectations:  None.
>>> Meaning of word letter variable |letter!56| is letter


Word received:  the

Attach:
        (det (bar 0) the
          (Expects:  (noun (bar 2))))

Call hypothesizers:   hypoth1

Votes returned:
  Vote number 1 by hypoth1 (detects expectations)
    Action:  (attach-new-word @stack 0 0)
    Time:  1
    Goodness:  100

Winning vote:  vote number 1
Result:
  *** Node printed twice because it is still open. ***
  (det (bar 2)
    (det (bar 1)
      (det (bar 0) the))
```

```
  (Expects:  (noun (bar 2) at det (bar 1))))

*** Node printed twice because it is still open. ***
(verb (bar 2)
  (verb (bar 1)
    (verb (bar 0) read)
    (det (bar 2)
      (det (bar 1)
        (det (bar 0) the))
      (Expects:  (noun (bar 2) at det (bar 1)))))
  (Expects:  ((optionally) det (bar 2) at verb (bar 1))))

*** Node printed twice because it is still open. ***
(infl (bar 1)
  (infl (bar 0) agr)
  (verb (bar 2)
    (verb (bar 1)
      (verb (bar 0) read)
      (det (bar 2)
        (det (bar 1)
          (det (bar 0) the))
        (Expects:  (noun (bar 2) at det (bar 1)))))
    (Expects:  ((optionally) det (bar 2) at verb (bar 1)))))

(infl (bar 2)
  (det (bar 2) mark)
  (infl (bar 1)
    (infl (bar 0) agr)
    (verb (bar 2)
      (verb (bar 1)
        (verb (bar 0) read)
        (det (bar 2)
          (det (bar 1)
            (det (bar 0) the))
          (Expects:  (noun (bar 2) at det (bar 1)))))
      (Expects:  ((optionally) det (bar 2) at verb (bar 1))))))


Morph expectations:  None.
Case expectations:  None.

Word received:  letter

Attach:
```

```
            (noun (bar 0) letter)

Call hypothesizers:  hypoth1 hypoth7

Votes returned:
  Vote number 1 by hypoth1 (detects expectations)
    Action:  (attach-new-word @stack 0 0)
    Time:  1
    Goodness:  100

Winning vote:  vote number 1
Result:
  (infl (bar 2)
    (det (bar 2) mark)
    (infl (bar 1)
      (infl (bar 0) agr)
      (verb (bar 2)
        (verb (bar 1)
          (verb (bar 0) read)
          (det (bar 2)
            (det (bar 1)
              (det (bar 0) the)
              (noun (bar 2)
                (noun (bar 1)
                  (noun (bar 0) letter))))))))


Morph expectations:  None.
Case expectations:  None.


Word received:  to

Attach:
        (unset (bar unset) to)

Call hypothesizers:  hypoth1 hypoth3 hypoth5 hypoth4

Votes returned:
  Vote number 1 by hypoth3 (detects possible post-modification)
    Action:  (attach-modifier @stack #<ANODE 12400154>
                (fix-meaning #<ALEAF 12400117> (quote goal)))
    Time:  4
    Goodness:  100
  Vote number 2 by hypoth3 (detects possible post-modification)
```

```
    Action:  (attach-modifier @stack #<ANODE 12400263>
                        (fix-meaning #<ALEAF 12400117> (quote goal)))
    Time:  11
    Goodness:  100
  Vote number 3 by hypoth3 (detects possible post-modification)
    Action:  (attach-modifier @stack #<ANODE 12400263>
                        (fix-meaning #<ALEAF 12400117> (quote purpose)))
    Time:  11
    Goodness:  100
  Vote number 4 by hypoth5 (detects possible argument/modifier)
    Action:  (attach-modifier @stack #<ANODE 12400263>
                        (fix-meaning #<ALEAF 12400117> (quote goal)))
    Time:  3
    Goodness:  100
  Vote number 5 by hypoth5 (detects possible argument/modifier)
    Action:  (attach-modifier @stack #<ANODE 12400263>
                        (fix-meaning #<ALEAF 12400117> (quote purpose)))
    Time:  5
    Goodness:  100
```

Note:

- *# < ANODE12400154 > is the projection of the noun "letter".*
- *# < ANODE12400263 > is the projection of the verb "read".*
- *# < ALEAF12400117 > is the word "to". "Purpose" is a meaning of "to" as an INFL; "Goal" is a meaning of "to" as a preposition.*

```
Winning vote:  vote number 4
Result:
  *** Node printed twice because it is still open. ***
  (prep (bar 2)
    (prep (bar 1)
      (prep (bar 0) to))
    (Expects:  (det (bar 2) at prep (bar 1))))

  (infl (bar 2)
    (det (bar 2) mark)
    (infl (bar 1)
      (infl (bar 0) agr)
      (verb (bar 2)
        (verb (bar 1)
          (verb (bar 1)
            (verb (bar 0) read)
            (det (bar 2)
              (det (bar 1)
```

```
                (det (bar 0) the)
                (noun (bar 2)
                  (noun (bar 1)
                    (noun (bar 0) letter))))))
          (prep (bar 2)
            (prep (bar 1)
              (prep (bar 0) to))
            (Expects:  (det (bar 2) at prep (bar 1))))))))


Morph expectations:  None.
Case expectations:  None.
Moving the stack boundry.


Word received:  chrysanne

Attach:
        (det (bar 2) chrysanne)

Call hypothesizers:   hypoth1

Votes returned:
  Vote number 1 by hypoth1 (detects expectations)
    Action:  (attach-new-word @stack 0 0)
    Time:  1
    Goodness:  100

Winning vote:  vote number 1
Result:
  *** Node printed twice because it is still open. ***
  (prep (bar 2)
    (prep (bar 1)
      (prep (bar 0) to)
      (det (bar 2) chrysanne)))

=======stack boundry=======

  (infl (bar 2)
    (det (bar 2) mark)
    (infl (bar 1)
      (infl (bar 0) agr)
      (verb (bar 2)
        (verb (bar 1)
          (verb (bar 1)
```

```
              (verb (bar 0) read)
              (det (bar 2)
                (det (bar 1)
                  (det (bar 0) the)
                  (noun (bar 2)
                    (noun (bar 1)
                      (noun (bar 0) letter))))))
            (prep (bar 2)
              (prep (bar 1)
                (prep (bar 0) to)
                (det (bar 2) chrysanne)))))))


Morph expectations:  None.
Case expectations:  None.


Word received:
Moving the stack boundry.

End sentence:
    Make completed sentence a CP.

Attach:
        (unset (bar 0) nullcomp)

The buffer is empty, so we just add maximal projection to buffer.

Attach:
        (infl (bar 2)
          (det (bar 2) mark)
          (infl (bar 1)
            (infl (bar 0) agr)
            (verb (bar 2)
              (verb (bar 1)
                (verb (bar 1)
                  (verb (bar 0) read)
                  (det (bar 2)
                    (det (bar 1)
                      (det (bar 0) the)
                      (noun (bar 2)
                       (noun (bar 1)
                        (noun (bar 0) letter))))))
                (prep (bar 2)
                  (prep (bar 1)
```

```
                     (prep (bar 0) to)
                     (det (bar 2) chrysanne))))))

Call hypothesizers:   hypoth1 hypoth3 hypoth5

Votes returned:
  Vote number 1 by hypoth1 (detects expectations)
    Action:  (attach-new-word @stack 0 0)
    Time:  1
    Goodness:  100

Winning vote:  vote number 1
Result:
=======stack boundry=======

  (comp (bar 2)
    (comp (bar 1)
      (comp (bar 0) nullcomp)
      (infl (bar 2)
        (det (bar 2) mark)
        (infl (bar 1)
          (infl (bar 0) agr)
          (verb (bar 2)
            (verb (bar 1)
              (verb (bar 1)
                (verb (bar 0) read)
                (det (bar 2)
                  (det (bar 1)
                    (det (bar 0) the)
                    (noun (bar 2)
                      (noun (bar 1)
                       (noun (bar 0) letter))))))
              (prep (bar 2)
                (prep (bar 1)
                  (prep (bar 0) to)
                  (det (bar 2) chrysanne))))))))
```

Chapter 5

Conclusion

5.1 Contributions

The goal of this thesis has been to propose a psychologically plausible model of parsing that would be interesting to both linguists and computer scientists. We identified several important psychological claims about the human sentence parsing mechanism, namely that processing is influenced by limitations on working memory and by a number of structural preferences, such as Right Association, Minimal Attachment, Revision as Last Resort, and verb-frame preferences, and then saw how some existing sentence processing systems deal with these issues. What we found was that although many of them give reasonable accounts of some subset of the psycholinguistic data, none provide a good account of how the remaining data could be integrated consistently. Then we proposed a processing model that tries to give a reasonable account of how one might bring all these necessary contraints and preferences together in one system. The starting point for the proposal was the Sausage Machine model (Frazier and Fodor, 1978; Fodor and Frazier, 1980), which provides a good account of memory constraints and sentence complexity, and incorporates most of the structural preferences we were seeking to include. From there we attempted to overcome the more serious deficiencies of the Sausage Machine model, namely its dependence on unprincipled aspects of its grammar, and its omission of verb-frame preferences. This thesis thus proposed the following modifications, and described how these modifications might be made:

- Substitute a principled theory of grammar, such as Brunson's interpretation of Government-Binding theory.
- Use estimated timing information to resolve conflicting preferences, *i.e.*, activate a set of processors (called "hypothesizers") to propose attachments, and execute the attachment that is successfully identified first.
- Add mechanisms to handle lexical disambiguation and semantic processing in parallel with syntactic processing.

The result is a sentence processor that uses a principled, well-known, and widely respected theory of grammar, that also acknowledges the existence of ambiguity and human preferences for resolving ambiguity, and that provides a mechanism for integrating several accounts of the psychological data and for resolving conflicts among them. The sentence processor also provides new support for the original Sausage Machine model by demonstrating that its deficiencies are apparently surmountable.

5.2 Some Needed Work

The model as proposed and implemented has several deficiencies of its own. Some of these deficiencies were discussed in section 3.4, namely the need for a better morphological processor and mechanisms for detecting and filling gaps, and resolving anaphoric references. In addition, more work must be done on designing additional hypothesizers for proposing more intermediate structure and revision, and a mechanism for checking semantic well-formedness and triggering reanalysis, if necessary, must be added. The semantic interpretation process could also use some refinement (see Hirst (1987) for a discussion of what Absity does not do).

5.3 Some Open Questions for Psychology

The proposed model also leaves open some questions for psychological research, including how we might verify the model, and what progress we could make toward refining processor speeds.

5.3.1 Verifying the Model

Because of the flexibility of the model presented here, it would be very difficult to prove or disprove its validity simply on the basis of the results of any particular parameterization of the individual timing functions. Verification would require that one first obtain some real data on how long certain processes (such as accessing thematic knowledge) take, adjusting the timing functions to fit the real data, and then checking to see if the resulting structural decisions match the observed preferences of the human sentence parsing mechanism.

5.3.2 Refining Processor Speeds

If, despite the difficulty in verification, we accept the model as correct because it gives a reasonable account of many psycholinguistic constraints and preferences already widely accepted, one can still make some progress in determining what the correct timing functions *should* look like. One way to estimate relative speeds would be to look at the frequency with which a particular type of attachment preference determines the preferred structure of sentences. Ford et al. (1982), suggested a similar approach to determining the strength of lexical preferences. Recent work in determining the relative strength of cues people use to fill linguistic roles confirms the validity of this suggestion. McDonald (1987) looks at linguistic cues such as word order, noun-verb agreement, and selectional properties (*e.g.* animacy). She found that her statistical analysis of "conflict validities" (the frequency with which a cue dominates conflicting cues) allowed her to make very good predictions about role assignments. Thus, it might be helpful to do a statistical analysis of overall validities and conflict validities for preferred sentence structures. Conflict validities should tell us what the timing functions should look like relative to one another. Unfortunately, conflict validities for sentence processing strategies may be difficult to determine accurately, because of the difficulty in finding relevant examples. Also note that this kind of analysis will not reveal online phenomena, although it would seem reasonable that the HSPM would learn these validities and change its weights so that the right responses would get activated faster. Judgements of meaning necessary to compute these validities will necessarily be subjective, although the nature of validities probably makes it sufficient to look at eventual interpretations in actual contexts, which would make subjective questionnaires or consultations with experts more dependable.

Appendix A

Correspondence between Syntactic and Semantic Types

Syntactic Type	**Semantic Type**
Adjective	sf-pair
$\overline{A}$	sf-pair
AP	sf-pair
Complementizer	frame-determiner
$\overline{C}$	frame-statement
CP	frame-statement
Determiner	frame-determiner
$\overline{D}$	frame-statement
DP	frame-statement
INFL	f-infl
$\overline{I}$	frame-statement/frame-determiner
IP	frame-statement/frame-determiner
Noun	frame
$\overline{N}$	frame-descriptor
NP	frame-descriptor
Preposition	slot
$\overline{P}$	sf-pair
PP	sf-pair
Verb	frame
$\overline{V}$	frame-descriptor
VP	frame-descriptor

References

Aaronson, Doris and Ferres, Steven (1986). Reading strategies for children and adults: A quantitative model. *Psychological Review*, 93(1):89–112.

Abney, Stephen and Cole, Jennifer (1985). A government-binding parser. In *Proceedings of the Fifteenth Annual Meeting of the North Eastern Linguistic Society.*

Abney, Steven Paul (1987). *The English Noun Phrase in Its Sentential Aspect.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Arens, Yigal, Granacki, John J., and Parker, Alice C. (1987). Phrasal analysis of long noun sequences. In *The 25th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, Stanford. pages 59–64.

Baddeley, Alan D. (1981). The concept of working memory: A view of its current state and probable future. *Cognition*, 10:17–23.

Baddeley, Alan D. and Hitch, G. (1974). Working memory. In Bower, G. H., editor, *The Psychology of Learning and Motivation*, volume 8. Academic Press, New York, pages 47–89.

Becker, Sue and Greiner, Russ (1987). On the implementation of a frame language in MRS. Manuscript, Artificial Intelligence Laboratory, University of Toronto.

Bever, Thomas G., Carrithers, Caroline, and Townsend, David J. (1987). A tail of two brains –or– the sinistral quasimodality of language. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, Washington. pages 764–773.

Blumenthal, Arthur L. (1966). Observations with self-embedded sentences. *Psychonomic Science*, 6(10):453–454.

Brunson, Barbara Anne (1986). Thematic argument structure. manuscript.

Brunson, Barbara Anne (1988). A processing model for Walpiri syntax and implications for linguistic theory. Technical Report 208, Computer Systems Research Institute, University of Toronto. Based on Forum Paper, Department of Linguistics, University of Toronto, August 1986.

Carlson, Greg N. and Tanenhaus, Michael K. (1987). Thematic roles and language comprehension. In Wilkins, Wendy, editor, *Thematic Structure.*

Carroll, John M., Tanenhaus, Michael K., and Bever, Thomas G. (1978). The perception of relations: The interaction of structural, functional, and contextual factors in the segmentation of sentences. In Levelt, Willem J. M. and Flores d'Arcais, G. B., editors, *Studies in the Perception of Language.* John Wiley & Sons, Chichester, pages 219–246.

Charniak, Eugene (1983). A parser with something for everyone. In King, Margaret, editor, *Parsing Natural Language.* Academic Press, London, pages 117–150.

Chomsky, Noam (1981). *Lectures on Government and Binding.* Foris, Dordrecht.

Church, Kenneth Ward (1980). On memory limitations in natural language processing. M.Sc. Thesis MIT/LCS/TR-245, Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge.

Clifton, Charles, Frazier, Lyn, and Connine, Cynthia (1984). Lexical expectations in sentence comprehension. *Journal of Verbal Learning and Verbal Behavior*, 23(6):696–708.

Connine, Cynthia, Ferreira, Fernanda, Jones, Charlie, Clifton, Jr., Charles, and Frazier, Lyn (1984). Verb frame preferences: Descriptive norms. *Journal of Psycholinguistic Research*, 13(4):307–319.

Cottrell, Garrison Weeks (1988). *A Connectionist Approach to Word Sense Disambiguation.* Morgan Kaufmann Publishers, Palo Alto, CA.

Cowper, Elizabeth A. (1987). An introduction to syntactic theory: The government-binding approach. Manuscript, University of Toronto, Department of Linguistics.

Crain, Stephen and Steedman, Mark (1985). On not being led up the garden path: The use of context by the psychological syntax processor. In Dowty, D. R., Karttunen, L., and Zwicky, A. M., editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives.* Cambridge University Press, pages 320–358.

Daneman, Meredyth and Carpenter, Patricia (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, 19:450–466.

Di Scuillo, Anna-Maria and Williams, Edwin (1987). *On the Definition of Word.* The MIT Press, Cambridge, MA. Linguistic Inquiry monograph 14.

Dorr, Bonnie (1987). A principle-based approach to parsing for machine translation. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, Washington. pages 78–88.

Dowty, David, Wall, Robert E., and Peters, Stanley, editors (1981). *Introduction to Montague Semantics.* D. Reidel, Dordrecht.

Ebbinghaus, H. (1885). *Über das Gedächtnis.* Duncker & Humbot, Leipzig.

Ferreira, Fernanda (1986). The role of context in resolving syntactic ambiguity. *University of Massachusetts Occasional Papers in Linguistics*, 9:40–62.

Ferreira, Fernanda and Clifton, Jr., Charles (1986). The independence of syntactic processing. *Journal of Memory and Language*, 25:348–368.

Fink, Stephen (1978). Case grammar and valence theory at a stalemate? Their relevance for semantic memory. In Abraham, Werner, editor, *Valence, Semantic Case and Grammatical Relations.* John Benjamins B. V., Amsterdam, pages 177–190.

Fodor, Janet Dean (1978). Parsing strategies and constraints on transformations. *Linguistic Inquiry*, 9(3):427–473.

Fodor, Janet Dean and Frazier, Lyn (1980). Is the human sentence parsing mechanism an ATN? *Cognition*, 8:417–459.

Ford, Marilyn, Bresnan, Joan, and Kaplan, Ronald (1982). A competence-based theory of syntactic closure. In Bresnan, Joan, editor, *The Mental Representation of Grammatical Relations.* The MIT Press, pages 727–796.

Frazier, Lyn (1978). *On Comprehending Sentences: Syntactic Parsing Strategies.* Indiana University Linguistics Club, Bloomington.

Frazier, Lyn and Fodor, Janet Dean (1978). The Sausage Machine: A new two-stage parsing model. *Cognition*, 6:291–325.

Frazier, Lyn and Rayner, Keith (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14:178–210.

Frazier, Lyn, Taft, Lori, Roeper, Tom, Clifton, Jr., Charles, and Ehrlich, Kate (1984). Parallel structure: A source of facilitation in sentence comprehension. *Memory and Cognition*, 12(5):421–430.

Gorrell, Paul Griffith (1987). *Studies of Human Syntactic Processing: Ranked-Parallel Versus Serial Models.* PhD thesis, University of Connecticut.

Guindon, Raymonde, Shuldberg, Kelly, and Conner, Joyce (1987). Grammatical and ungrammatical structures in user-advisor dialogues: Evidence for sufficiency of restricted languages in natural language interfaces to advisory systems. In *The 25th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, Stanford. pages 41–44.

Hirst, Graeme (1983). A foundation for semantic interpretation. In *The 21st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, Cambridge, MA. pages 64–73.

Hirst, Graeme (1984). Jumping to conclusions: Psychological reality and unreality in a word disambiguation program. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, CO. pages 179–182.

Hirst, Graeme (1987). *Semantic Interpretation and the Resolution of Ambiguity.* Cambridge University Press, Cambridge.

Hirst, Graeme (1988). Resolving lexical ambiguity computationally with spreading activation and Polaroid Words. In Small, Steven, Cottrell, Garrison, and Tanenhaus, Michael, editors, *Lexical Ambiguity Resolution.* Morgan Kaufmann Publishers, San Mateo, CA.

Hirst, Graeme and Charniak, Eugene (1982). Word sense and case slot disambiguation. In *Proceedings, National Conference on Artificial Intelligence (AAAI-82)*, Pittsburgh. pages 95–98.

Holmes, Virginia M. (1984). Parsing strategies and discourse context. *Journal of Psycholinguistic Research*, 13(3):237–257.

Huang, Xiuming (1985). Machine translation in the SDCG formalism. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, New York.

Huang, Xiuming and Guthrie, Louise (1985). Parsing in parallel. Technical Report MSSC-85-40, Computing Research Laboratory, New Mexico State University, Las Cruces, NM.

Hudson, Susan B. and Tanenhaus, Michael K. (1985). Phonological code activation during listening. *Journal of Psycholinguistic Research*, 14(6):557–567.

Jackendoff, Ray (1972). *Semantic Interpretation in Generative Grammar.* The MIT Press, Cambridge, Mass.

Jackendoff, Ray (1987). The status of thematic relations in linguistic theory. *Linguistic Inquiry*, 18(3):369–411.

Kimball, John P. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, 2:15–47.

Klatsky, Roberta L. (1980). *Human Memory: Structures and Processes.* W. H. Freeman and Company, New York, 2nd edition.

Levelt, Willem J. M. (1978). A survey of studies in sentence perception: 1970–1976. In Levelt, Willem J. M. and Flores d' Arcais, G. B., editors, *Studies in the Perception of Language.* John Wiley & Sons, Chichester.

Levesque, Hector and Mylopoulos, John (1979). A procedural semantics for semantic networks. In Findler, Nicholas V., editor, *Associative Networks: Representation and Use of Knowledge by Computers.* Academic Press, pages 93–120.

Macdonald, Antonina Hansell (1979). *The Influence of Duration on the Processing of English Sentences.* PhD thesis, The University of Michigan.

Marcus, Mitchell P. (1980). *A Theory of Syntactic Recognition for Natural Language.* The MIT Press, Cambridge, MA.

Marcus, Mitchell P., Hindle, Donald, and Fleck, Margaret M. (1984). D-theory: Talking about talking about trees. In *The 21st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference.* pages 129–136.

Marslen-Wilson, William (1975). Sentence perception as an interactive parallel process. *Science*, 189(4198):226–228.

Marslen-Wilson, William and Tyler, Lorraine K. (1980). The temporal structure of spoken language understanding. *Cognition*, 8:1–71.

Marslen-Wilson, William D., Tyler, Lorraine K., and Seidenberg, Mark (1978). Sentence processing and the clause boundry. In Levelt, Willem J. M. and Flores d' Arcais, G. B., editors, *Studies in the Perception of Language.* John Wiley & Sons, Chichester, pages 219–246.

McDonald, Janet L. (1987). Assigning linguistic roles: The influence of conflicting cues. *Journal of Memory and Language*, 26:100–117.

Miller, George A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97.

Milne, Robert W. (1982). Predicting garden path sentences. *Cognitive Science*, 6:349–373.

Mitchell, D. C. and Holmes, Virginia. M. (1985). The role of specific information about the verb in parsing sentences with local structural ambiguity. *Journal of Memory and Language*, 24:542–559.

Montague, Richard (1973). The proper treatment of quantification in ordinary English. In Hintikka, Kaarlo Jaakko Juhani, Moravcsik, Julius Matthew Emil, and Suppes, Patrick Colonel, editors, *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics.* D. Reidel, Dordrecht, pages 221–242.

Moyne, John (1985). *Understanding Language: Man or Machine.* Plenum Press.

Pulman, S. G. (1986). Grammars, parsers and memory limitations. *Language and Cognitive Processes*, 1(3):197–225.

Rayner, Keith, Carlson, Marcia, and Frazier, Lyn (1983). The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior*, 22:358–374.

Riesbeck, Christopher and Schank, Roger (1978). Comprehension by computer: Expectation-based analysis of sentences in context. In Levelt, Willem J. M. and Flores d' Arcais, G. B., editors, *Studies in the Perception of Language.* John Wiley & Sons, Chichester, pages 247–294.

Russell, Stuart (1985). The compleat guide to MRS. Technical Report KSL-85-12, Stanford Knowledge Systems Laboratory, Stanford University.

Schubert, Lenhart (1984). On parsing preferences. In *Coling 84 Proceedings*. pages 247–250.

Schubert, Lenhart (1986). Are there preference trade-offs in attachment decisions? In *Proceedings, National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia. pages 601–605.

Sells, Peter (1985). *Lectures on Contemporary Syntactic Theories: An Introduction to Government-Binding Theory, Generalized Phrase Structure Grammar, and Lexical-Functional Grammar*. Center for the Study of Language and Information, Stanford University.

Shieber, Stuart M. (1983). Sentence disambiguation by a shift-reduce parsing technique. In *The 21st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, Cambridge, MA. pages 113–118.

Small, Stephen L. (1980). *Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding*. PhD thesis, University of Maryland. Department of Computer Science.

Somers, Harold L. (1987). *Valency and Case in Computational Linguistics*. Edinburgh University Press. Edinburgh information technology series 3.

Underwood, Geoffrey (1985). Eye movements during the comprehension of written language. In Ellis, Andrew W., editor, *Progress in the Psychology of Language, Volume 2*. Lawrence Erlbaum Associates, London, pages 45–71.

Vater, Heinz (1978). On the possibility of distinguishing between complements and adjuncts. In Abraham, Werner, editor, *Valence, Semantic Case and Grammatical Relations*. John Benjamins B. V., Amsterdam, pages 21–45.

Von Eckardt, Barbara and Potter, Mary C. (1985). Clauses and the semantic representation of words. *Memory and Cognition*, 13(4):371–376.

Wanner, Eric (1980). The ATN and the Sausage Machine: Which one is baloney? *Cognition*, 8:209–225.

Weber, Robert J., Burt, Diana Byrd, and Noll, Nicholas C. (1986). Attention switching between perception and memory. *Memory and Cognition*, 14(3):238–245.

Wehrli, Eric (1984). A government-binding parser for french. Working paper no. 48, Institut pour les études sémantiques et cognitives, Université de Genève.

Weinberg, Amy (1987). Modularity in the syntactic parser. In *Modularity in Knowledge Representation and Natural-Language Understanding*. The MIT Press, Cambridge, MA, pages 259–276.

Wilks, Yorick, Huang, Xiuming, and Fass, Dan (1985). Syntax, preference, and right attachment. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles. pages 779–784.

Wingfield, Arthur and Nolan, Karen A. (1980). Spontaneous segmentation in normal and in time-compressed speech. *Perception and Psychophysics*, 28(2):97–102.

Woods, William Aaron (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, 13:591–602.

Zhang, Guojun and Simon, Herbert A. (1985). STM capacity for Chinese words and idioms: Chunking and acoustical loop hypotheses. *Memory and Cognition*, 13(3):193–201.