Mechanisms for Sentence Processing

Matthew W. Crocker
Centre for Cognitive Science
The University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW
UK
mwc@cogsci.ed.ac.uk

July 6, 1998

1 Introduction

Sentence processing is the means by which the words of an utterance are combined to yield the interpretation of a sentence. It is a task which all people can do well: quickly, efficiently, effortlessly, and accurately. Unlike solving calculus problems or playing chess, we can hardly fail to be successful at understanding language. This wouldn't be surprising, were it not for the fact that language is not only extremely complex, but also highly ambiguous.

An interesting contrast can be made between lexical access and sentence processing. In the case of lexical access, we might simply imagine that the problem is one of matching the phonological or orthographic features of an input word or morpheme with an entry in our mental lexicon. Naively, this is simply a process of using the input as a key into a large database. This is simplifying things a great deal of course, but if we contrast this with sentence processing, we can quickly see that something very different is going on in the latter. While you have seen all the words in the previous sentence before, you have probably never seen the sentence itself before. Thus we can easily imagine how one might retrieve the meanings of the individual words, but how does one understand the meaning of the sentence as a whole? You presumably have never seen a sentence with precisely that meaning before, yet you arrived at the intended interpretation effortlessly and immediately.

This chapter is concerned with how the task of sentence processing is accomplished, considering first the basic issues and then examining a number

of proposals which have been advanced. In particular, the study of sentence processing touches on many of the interesting issues facing cognitive science more generally. These include: the nature of mental *representations*; the *algorithms* which are used to construct them; the extent to which the sentence processor is distinct or *modular*; what factors lead to increased processing *complexity*; what *strategies* are adopted; and why?

In our quest for a model of how people process the utterances they hear, we need first to consider two fundamental sources of information: an appropriate formal description of how the words of a sentence can be structured into a connected, interpretable representation, often characterised by a grammar, and also empirical evidence concerning people's behaviour when they process language. In this chapter we will provide a brief overview of grammar rules and representations, and then consider what kinds of mechanisms might be used to build syntactic representations, or *analyses*, using such grammatical knowledge, with the aim of modelling the human sentence parsing mechanism (HSPM).

We will begin with a general discussion of the relationship between the grammar and a parser, since much of what follows relies on a clear understanding these underlying topics. We then provide an overview of natural language ambiguity, a phenomenon which plagues the construction of artificial natural language parsing systems, while providing an important window into the nature of the human sentence processor. Finally, we turn our attention to the specific kinds of mechanisms which might be used to characterise the human sentence processor. Here we consider a range of parsing algorithms and strategies for resolving ambiguity, and attempt to evaluate them with respect to their explanation of the data, their plausibility as psychological models, and their ability to make clear predictions.

1.1 Grammars and Parsers

In building the representations of an utterance which allow us to interpret language, the sentence processor must bridge the automatic perceptual task of word recognition and lexical access with the more conscious, inferential processes of language understanding. Perhaps the most important aspect of sentence processing is that it is compositional; that is to say, the interpretation of a sentence is determined by the particular way its words can be combined. As we have noted already, this interpretation cannot simply be retrieved from our memory of previous sentences we have encountered, otherwise we would not be able to understand the novel utterances which make up most of our linguistic experience. Rather, an interpretation must

be constructed when each new utterance is heard or read. Furthermore, current linguistic theories would suggest that it is indeed a very rich set of constraints, rules, representations and principles which determine how words may be combined to yield structures which can be interpreted appropriately.

Linguistic theories have been developed with precisely the aim of (a) licensing for those utterances which are part of the language under investigation, and ruling out those which are not and (b) providing an analysis of well-formed utterances which can then serve as the basis for semantic interpretation.[1] To construct a syntactic analysis for a particular sentence, the linguist must show that the rules of grammar can be used to derive, or *generate*, the utterance in question. If there is no such analysis, then the grammar does not generate the utterance, and it is considered to be ungrammatical with respect to the grammar. Much work in computational linguistics has been devoted to the development of algorithms which can automatically construct such a syntactic analysis for a particular sentence, or *parse* it, given a particular grammar. As this corresponds closely to the task of the human sentence processor, as sketched above, it is useful to consider how parsers and grammars have been formalised within (computational) linguistics.

S	$\rightarrow$	NP VP	Det	$\rightarrow$	{*the, a, every*}
NP	$\rightarrow$	PN	N	$\rightarrow$	{*man, woman, book, hill, telescope*}
NP	$\rightarrow$	Det N	PN	$\rightarrow$	{*John, Mary*}
NP	$\rightarrow$	NP PP	P	$\rightarrow$	{*on, with*}
PP	$\rightarrow$	P NP	V	$\rightarrow$	{*saw, put, open, read, reads*}
VP	$\rightarrow$	V			
VP	$\rightarrow$	V NP			
VP	$\rightarrow$	V NP PP			

Figure 1: A Simple Phrase Structure Grammar

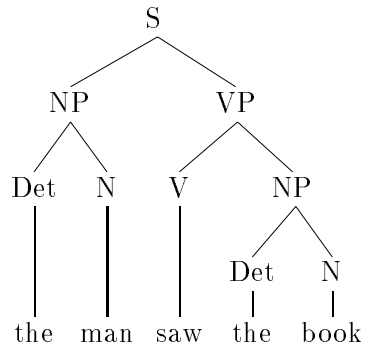
To make our discussion more concrete, consider the phrase structure grammar and lexicon in Figure 1, which covers a tiny fragment of English. The language defined by this grammar is, by definition, the complete set of sentences that are grammatical according to (or *generated* by) the rules provided. So, for example,

(1) "the man saw the book."

[1] This is typically referred to as *descriptive adequacy*. In addition linguistic theories often seek to be *explanatory*, to the extent that they can shed light on, for example, how language might be acquired.

is in the language, as shown by the following phrase structure derivation:

(2)



However, the following two sentences are excluded by this particular grammar:

(3) (a) "the man saw the film."
(b) "the man saw the woman open the book."

Sentence (3a) is quite clearly out, for the simple reason that the word *film* is not in the lexicon, and (3b) because there is no derivation using the phrase structure rules in the given grammar. Despite the fact that this particular grammar is tiny, and not very representative of English, it is important to note that the language defined by this grammar is in fact infinite. This is because of the *recursive* rule $NP \longrightarrow NP\ PP$, which can in principle be used any number of times in generating a sentence, to produce sentences of arbitrary length (though these would be implausibly repetitive given the limited size of this grammar and lexicon). In this way, grammars of this sort represent a very powerful device, which have the potential to analyse, and therefore support the interpretation of, an infinite set of sentences.[2]

1.2 Ambiguity

Utterances may have more than one interpretation, and there can be several reason for such ambiguities. If we first consider lexical ambiguity, we note that a given word may exhibit both semantic and part-of-speech ambiguities:

[2]Closer inspection of the grammar will also reveal that while it only covers a tiny fragment of English, it still generates what we would consider ungrammatical sentences. For example, *The woman put* is permitted since, among other things, the grammar fails to account for the specific requirements of particular verbs (e.g. *put* must be followed by an NP and a PP).

(4) a. "I robbed the *bank*."
b. "I fished from the *bank*."
c. "I *bank* with Lloyds Bank."
d. "*Bank* the plane to turn it."

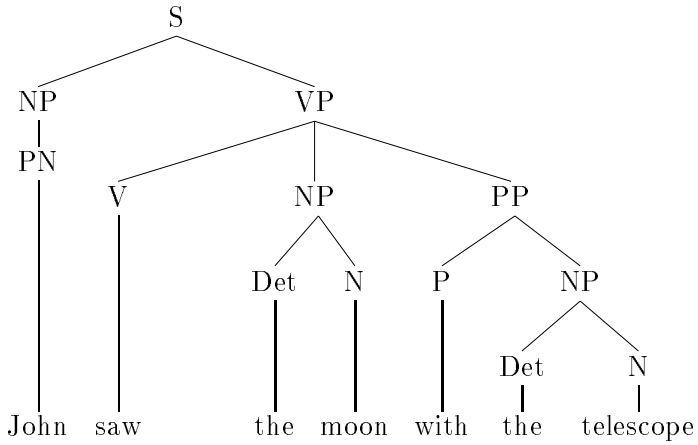
Sentences (3a&b) above, demonstrate how *bank* is used as a noun, but with two distinct meanings; a semantic 'sense' ambiguity. In (3c), *bank* is a verb, but has a meaning clearly related to that in (3a), but completely unrelated to (3b). Finally, (3d) demonstrates another verbal sense of *bank*, which is quite separate from the other uses (although possibly vaguely related to that in (3b)), partially illustrating the range of possible category and sense ambiguities which can occur in the lexicon.

Another possibility, of particular interest to the present discussion, is that sentences may have more that one possible grammatical structure associated with them. These are typically called structural, or syntactic, ambiguities. Consider, for example, the following sentence:

(5) "John saw the moon with the telescope."

In this sentence, the prepositional phrase *with the telescope* might be considered a modifier of *the moon* (i.e. if there were a telescope on the moon), but it seems more likely that it is the instrument of *saw*, and therefore modifies the verb phrase, as shown in the following parse tree:

(6)

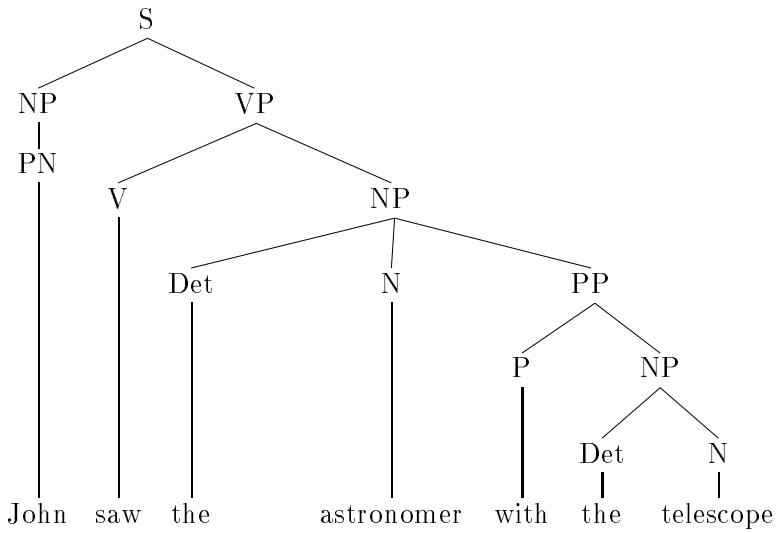


By changing just one word, however, the alternative structure becomes much more plausible (though not necessarily preferred). Consider,

(7) "John saw the astronomer with the telescope."

In this sentence, the prepositional phrase may well be attached to the *the astronomer*, since it is reasonable that there may be several astronomers, and the sentence is referring to the one who owns a telescope. If this is case, since it is *the astronomer* that's being modified, the parse tree would look as follows:

(8)



In the two examples above, the sentences have two possible interpretations, and are said to be *globally* ambiguous. It is also possible, if we accept that sentences are processed incrementally and left-to-right, that we might encounter *local* ambiguities. That is, situations where there are a number of possible analyses for the current, initial sub-string of the utterance, but which are disambiguated by the end of the sentence, when the entire string has been processed.

(9) "I knew the solution to the problem was incorrect."

In this sentence, the first ambiguity occurs as a result of *knew* being ambiguous as to the category of its complement: it may take either a noun phrase or a sentence. Thus, when the noun phrase *the solution to the problem* is encountered, we have two possible analyses: we can attach it directly as the object of *knew* or we can create a sentential complement with *the solution to the problem* as the embedded subject. In contrast with (7), however, the ambiguity is a temporary one, as only the sentential complement analysis is sustained when the remaining words *was incorrect* are encountered.

In fact, such sentences are only ambiguous if we assume *incremental* interpretation. Experimental evidence has demonstrated people do indeed

have mild, but systematic, difficulty with such sentences (Frazier & Rayner, 1982), suggesting that they do analyse the sentence incrementally, and typically adopt the direct object analysis initially. Changing to the embedded sentence then causes an increase in reading complexity. This sort of effect is made even clearer by sentences of the following sort:

(10) "The woman sent the letter was pleased"

If we again assume incremental processing, the most natural analysis to pursue for this sentence is to parse *sent* as the main verb, and *the letter* as its direct object. However the end of the sentence, *was pleased*, then leaves us with no way to continue. The verb *sent*, however, is ambiguous as either the simple past tense form or the past participle, so in fact the correct analysis is to parse *sent the letter* as a *reduced relative* clause, and *was pleased* as the main verb (cf. *The woman who was sent the letter was pleased*). Typically, people are unable to recover this alternative reading, and find the sentence to be ungrammatical unless the context is sufficiently biasing (cf. §4.1.2). Such examples of local ambiguities are often referred to as conscious 'garden paths', since they have the apparent effect of leading the human parser towards one, ultimately wrong, analysis from which it is difficult or impossible to recover.

1.3 The Competence Hypothesis

We have outlined above how grammars can be used to assign structural descriptions to sentences of a language, indeed, sometimes more than one if a particular sentence is ambiguous. We have tacitly assumed, here, that the sort of grammar proposed by linguists, similar in spirit to the one in Figure 1 though much more complex, approximates the knowledge of language, or *competence*, that we as humans possess, and apply in our everyday use of language, our *performance*. This might not be the case. It is entirely possible that people make use of a completely different grammar. For example, while linguists typically pursue the simplest, most elegant theories about language which capture as many generalisations and abstractions as possible, it may be the case that people make use of more specific grammatical knowledge tailored to cope more effectively with, say, frequent expressions and indeed certain 'ungrammatical' sentences.

It is, however, commonly assumed by researchers that people do make use of grammatical knowledge which is approximated by our current theories of grammar, or minimally, that people recover equivalent representations:

this assumption is referred to as the *Competence Hypothesis* and it will be assumed throughout this chapter (see Berwick & Weinberg (1984) for a more thorough technical discussion of this point). Such an assumption is essential if psycholinguistics is to shed light on the nature of language processes, since the linguistic theory is used to make predictions and shape the development of parsing models which can be tested empirically. That is, formal linguistic theory provides the common representational vocabulary linking computational and experimental psycholinguistics.

In the rest of this chapter we will consider two basic topics. The first concerns the range of possible mechanisms for constructing syntactic analyses, like the parse trees above. Secondly, we consider how people appear to cope with this problem of ambiguity, when constructing such a parse for the utterances they hear.

2 Parsing Issues

If we assume that a theory of grammar approximates people's 'knowledge' of their language's structure, then a theory of parsing must explain how people use that knowledge to construct an analysis or interpretation for the utterances they encounter. As mentioned above, research in computational linguistics has led to numerous possible techniques and strategies for parsing natural language. Much of the theory behind these results has its origins in the study of formal and computer languages (see Aho, Sethi, & Ullman (1986) for an overview), but other accounts have developed specifically to address problems involved in processing natural language as well.

In this chapter, we begin with a general discussion of parsing techniques, and introduce some standard parsing algorithms. We then consider some more specific proposals in the context of human language processing. Our aim is to consider how well current computational models of human sentence processing fare in explaining human language processing behaviour. In particular we are concerned with the following issues:

1. To what extent do the models process sentences incrementally, word-by-word, in the manner in which people appear to process language?

2. How are structural ambiguities dealt with? Do parsing decisions reflect those of human processing?

3. Are increases in processing complexity predicted to occur in precisely those instances where people demonstrate processing difficulties?

2.1 Parsing Algorithms

The task of a parser is to examine the string of words of an input sentence, and assign that string a well-formed syntactic structure, given a particular grammar. Crucially, there may be more than one analysis, if the sentence is syntactically ambiguous (as seen above), or there may be no analysis, if the sentence is not a member of the language defined by the grammar being used by the parser. The parsing algorithm specifies the procedures which are to be used to find the syntactic structure, or *parse tree*, for an utterance, and there are a number of different dimensions of variation for such algorithms.

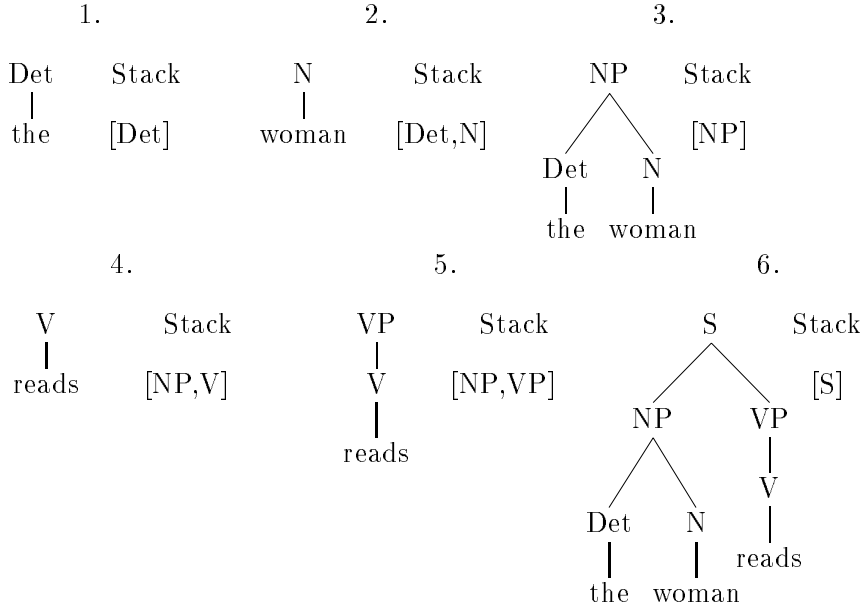
For example, the parser might work through the string from left-to-right, as people presumably do, or from right-to-left. Also, it may use the grammar to 'drive' the parsing process, first building structure and then matching it to the words of the string, or it may concentrate primarily on the words in the string, in a input-driven manner, and build structure 'bottom-up'. Algorithms also vary in how they handle ambiguities. Some, when faced with a situation where more than one structure could be built, will simply choose one. If that turns out to be incorrect, the parser will later 'backtrack' to the point where it made that choice (called a *choice-point*), and try an alternative. On the other hand, some parsers will try all possibilities in parallel, then when one or more of the parallel analyses fail, such a parse simply forgets about them, secure in the knowledge that the successful analysis (if it exists) is also being pursued.

In the following subsections we consider two possible parsing algorithms in greater detail. We will then go on to some issues of parsing *complexity* which have been used to defend particular algorithms as good approximations of the human parser.

2.1.1 Bottom-up

Let us begin by considering a parser which is principally driven by the words in the sentence, using the grammar in Figure 1 to combine them 'bottom-up', into higher level constituents. In this case we consider the 'shift-reduce' parser, which is the simplest instance of a bottom-up parser. It works by looking at the words of the sentence, and trying to combine them into constituents, using the rules of the grammar. As its name suggests there are two fundamental parsing operations: 'shift', which moves the algorithm to the next word in the sentence, and 'reduce', which tries to combine the constituents already found into new constituents. Below, we illustrate a simple bottom-up parse for the sentence *the woman reads*.

(11)



As constituents are built up, we use a 'stack' to keep track of what's been found so far. Thus, at step 1, we've found a determiner, so Det is 'pushed onto the stack'. At step 2, we've also found a noun, so N is also 'shifted' onto the stack, where the top of the stack is to the right. By keeping track of the categories found so far, the algorithm can then 'reduce' two categories on the top of the stack, if there is a rule in which these categories appear on the right-hand-side. So, at step 3, the Det and N are replaced by NP. That is, they are combined using the $NP \longrightarrow Det\ N$ rule. This continues until all the words have been processed, and the only category remaining on the stack is S, so we have found a sentence. The algorithm can be more explicitly stated as follows:

[12]

1. Initialise Stack = [ ] (empty)
2. Either *shift*:
   - Select the next word in the sentence (beginning with the first word).
   - Determine the category of the word in the lexicon
   - Push the category onto the top of the stack
3. Or *reduce*:
   - If the categories on the stack match those on the right-hand-side of any grammar rule, then:
     - Remove those categories from the stack
     - Push the category on the left-hand-side of the rule onto the top of the stack
4. If there are no more words in the sentence, then:
   - If the Stack = [S], then done.
5. Go to step 2.

To be clear about what this algorithm is saying, consider each of the 5 Steps. Step 1 simply initializes the algorithm's only data structure, the stack, to empty. That is, no constituents have been identified when the algorithm begins. Step 2 defines the *shift* operation, which simply looks at the next word in the sentence, beginning with the first word, moving left to right. It determines the category of the word, and pushes that category onto the top of the stack. For example in (11), step 1, the word *the* is identified as a Det, and the category Det is put on the stack. Step 3 defines the *reduce* operation, which sees if the top of the stack matches the right hand side of any rule. If so, those categories are removed from the stack and replaced by whatever category was on the left hand side of the rule. Thus in (11), step 3, the stack [Det,N] matches the right hand side of the $NP \longrightarrow Det\ N$, so those categories are removed from the stack and replaced by the category NP. Step 4 is the termination case, which states that if the algorithm has consumed all the input words and reduced them to the single distinguished symbol, then it has successfully parsed the sentence and the algorithm finishes. This is precisely the case in (11), step 6, when all the words have been consumed, and only the category S remains on the stack. If the algorithm is not done, then Step 5 causes the algorithm to continue by returning to Step 2.

It is important to note that Steps 2 & 3 are alternatives, and either may apply at any point, but not both, and neither has any priority. Within each

step there may also be alternatives, if (for Step 2) a particular word has more than one preterminal category (e.g. *bank* might be both a verb and a noun), or (for Step 3) more than one phrase structure rule has a right-hand-side that matches the categories on the top of the stack. This introduces an element of 'choice' or *non-determinism* into the algorithm, a matter to which we shall return in § 3. Furthermore, if at some point in parsing, the algorithm can neither *shift* nor *reduce*, then the algorithm has blocked. This will occur if the sentence is ungrammatical, or if it is grammatical but we made an incorrect choice at some point. In the latter case, some additional mechanism is required to permit the algorithm to pursue an alternative, and we will consider this issue in greater detail later.

2.1.2 Top-down

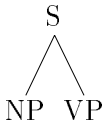
A top-down parser constructs a parse tree by first assuming that there is a sentence, and then working its way down the tree to the words themselves. That is, it begins by establishing S as the root of the tree, then finds a rule with S on the left hand side, and writes the categories on the right hand side as the daughters of S in the parse tree. This step is then repeated for the first daughter, and so on, until the bottom of the tree is reached (i.e. a word, or *terminal*, is parsed). Then the parser looks back up the tree, and to the right, until it finds another node which it can similarly expand. So, for the grammar given in (1), the first few steps in the parsing process for the sentence *The woman read the book* would be as follows:

(13)

1.

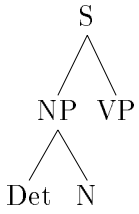
S	Stack
	[S]

2.



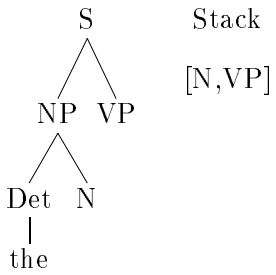
Stack
[NP,VP]

3.



Stack
[Det,N,VP]

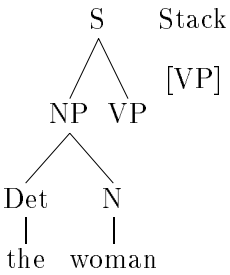
4.



Stack

[N,VP]

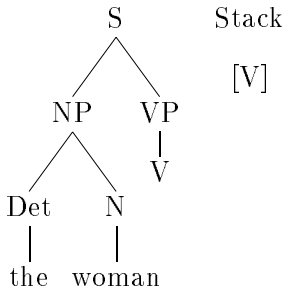
5.



Stack

[VP]

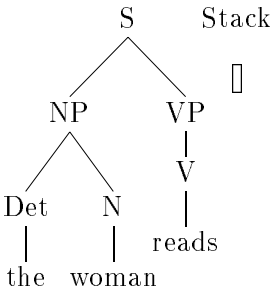
6.



Stack

[V]

7.



Stack

[]

The top-down algorithm can be expressed as follows:

[14]

---

1. Initialise Stack = [S]
2. If top element of the stack is a non-terminal N, then:
   - Select a rule which rewrites $N \longrightarrow R$ (where R is the symbol(s) on the right side of the rule).
   - Remove N from the stack
   - Add R to the top of the stack
3. If the top element of the stack is a pre-terminal P, then:
   - Find the next word W in the sentence
   - If there is a rule which rewrites $P \longrightarrow W$ then: remove the pre-terminal from the stack
   - Else fail
4. If Stack = [ ] (it's empty), and there are no more words to parse, then:
   - Succeed
5. Go to step 2

Once again, let us consider the steps involved in this algorithm. In contrast with the bottom-up parser, the top-down algorithm uses a stack

to keep track of categories which *need to be found*, rather than those which have *already been found*. As before, Step 1 initialises the stack, but since the algorithm is trying to find an S, the category S is put on the stack to represent this. Steps 2 & 3 constitute the 2 basic operations of the parser. Step 2 considers the case where the category on the top of the stack is a non-terminal (in fact, non-pre-terminal),[3] and finds a rule in the grammar which can expand this. In (13), step 2, for example, the category S is expanded using the rule $S \longrightarrow NP\ VP$. The result is that S is removed from the stack and replaced by [NP VP]. Step 3 handles the case where the category on the stack is a pre-terminal (i.e. must dominate a word). At this step the parser must ensure that there is a rule in the grammar which rewrites the pre-terminal category as the next word in the sentence, otherwise the algorithm blocks. This occurs in (13), step 5, when the pre-terminal N matches the input word 'woman' via the grammar rule $N \longrightarrow woman$. Step 4 is the termination case, which states that if no more categories remain to be found (the stack is empty), and there are no more words to be parsed, then the algorithm has successfully parsed the sentence and finishes. Otherwise, Step 5 causes the algorithm to continue by returning to Step 2.

As with the bottom-up algorithm, the top-down algorithm also has an element of non-determinism. This does not occur between Steps 2 & 3, since the category on the stack will always be either a non-terminal or a pre-terminal. Within Step 2, however, there may be more than one way to expand a particular non-terminal. For example, the category NP can expand as either Det, N, or PN depending on which grammar rule is used. We return to discussion of this in § 3.

2.2 Properties of the Parser

Let us suppose we wish to decide which of the two algorithms above most closely resembles the procedures that people use to interpret sentences. What criteria might we apply? We have already argued that one of the most salient properties of the human sentence parser is that it appears to operate incrementally. Both of the above algorithms are incremental in the sense that they process each word of the sentence one-by-one from left-to-right. But that only means they treat the *input* incrementally. If we assume, quite reasonably, that the *output* of the parser is a parse tree, and that this

[3] To clarify, *non-terminals* are categories which can dominate other categories, while *terminals* can only appear at the leaves of the tree and dominate nothing else. The notion of *pre-terminals* is used for categories such as N, V and P which stand for sets of terminals (i.e. words), and thus only ever dominate a terminal.

is to be used as the basis for subsequent semantic interpretation, then the two models differ substantially. In the case of the top-down parser there is a single 'connected' parse tree for the sentence after each word is processed (although it is not complete, until the end of the parse). In the bottom-up parser, however, adjacent constituents may be left on the stack for an arbitrarily long period. For example, the NP subject of a sentence will not be reduced with the VP (via the rule $S \longrightarrow NP\ VP$) until after the entire VP has been parsed (in the above example, the VP only contains the verb, but typically it will also contain a number of complements). Under the standard assumption that we cannot begin to evaluate the semantics and plausibility, until these constituents are connected in the parse tree, the bottom-up parser will lead to a psychologically implausible delay in interpretation for all sentences (see Stabler (1991) for discussion).

So from the standpoint of incrementality, the top-down parser fares much better than the bottom-up parser. The top-down parser, however, has problems of its own. To begin with, it attempts to construct large portions of the tree before even looking at the words in the sentence. In other words, given several (possibly dozens of) rules that expand a particular category, the parser simply has to choose one arbitrarily. Given that it makes such a guess at each node, the algorithm may fail numerous times before making the right sequence of guesses. The problem clearly is that the parser does not use the input to guide the decisions it makes. The bottom-up parser, on the other hand, is input-driven, but may leave large constituents sitting on the stack, and there fails to construct a single connected representation incrementally.

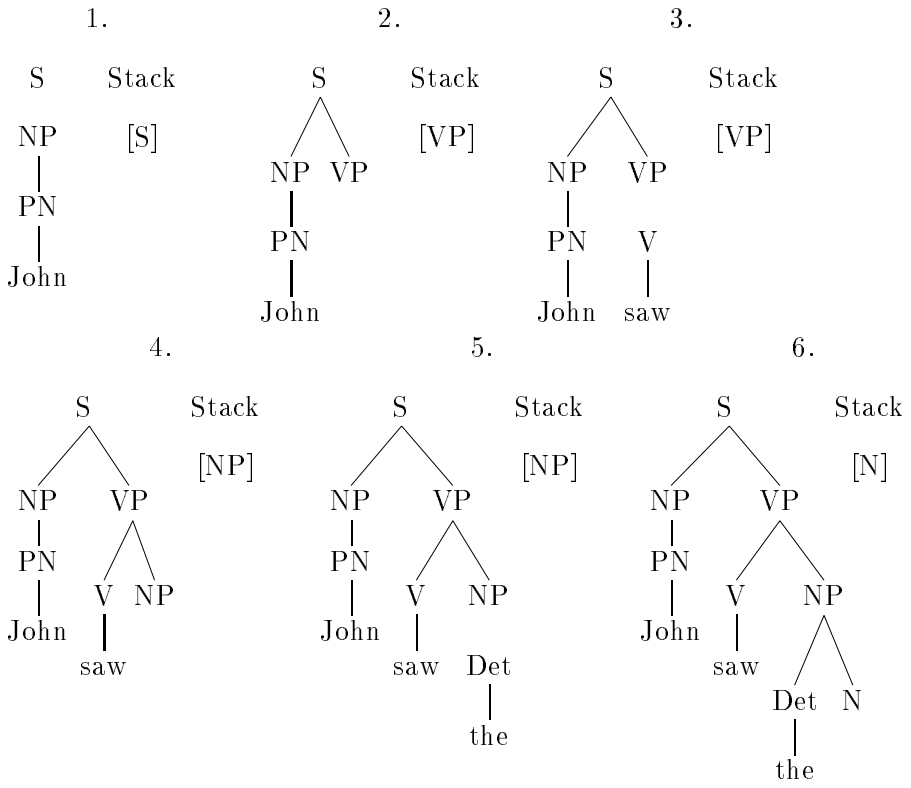
2.3 A Psychologically Plausible Parser: The Left-Corner Algorithm

The top-down and bottom-up algorithms represent two extremes of the vast range of possible parsing algorithms. One of the fundamental research goals in computational psycholinguistics has been to find the right parsing algorithm to meet the criteria of being both incremental and data-driven. An obvious strategy is to use a combined top-down/bottom-up algorithm. One now well-known instance of this is the 'left-corner' parsing algorithm.

The central intuition behind the left-corner algorithm is to use the 'left-corner' of a phrase structure rule (the left-most symbol on the right-hand side of the rule, i.e. the left-most daughter of a category), to project its mother category (the left-hand side of the rule), and predict the remaining categories on the right, top-down. Consider the parse sequence for the

sentence fragment *John saw the ...*:

(15)



In step 1, several things are happening: we make the top-down assumption that we are expecting to build a sentence, and S is placed on the stack. We also have found, bottom-up, a PN, and hence an NP. Step 2 illustrates the left-corner rule: given that we are looking for an S (as it is on the stack), and have found an NP, we can use the rule $S \longrightarrow NP\ VP$ to attach the NP as the left-corner of S, which is now removed from the stack, replaced by the VP we are now looking for. Step 3 simply shows that we have found a V, bottom-up, and in step 4 we similarly use the left-corner rule to attach V to VP, and predict an NP (via the rule $VP \longrightarrow V\ NP$). Thus VP is removed from the stack, replaced by newly predicted NP. We then continue by finding a Det, which is the left-corner of NP, and so on.

2.3.1 Incrementality

A quick inspection of the intermediate parse trees in (15) suggests that the algorithm builds up the tree incrementally, as each word is found. That is,

as each word is encountered, it projects its structure bottom-up, and then uses the left-corner strategy to attach it to the structure we are trying to build top-down. In fact, while this algorithm is highly incremental, it is not guaranteed to be. That is, given some grammars and input strings, the parser will delay building a completely connected structure. There are a number of variations of this algorithm which have pursued this problem in greater detail, and we will not digress further into the technical details here. The reader is referred to Shieber & Johnson (1993), Stabler (1994), and Crocker (1996) for more sophisticated discussion of related incremental parsing algorithms.

2.3.2 Memory Load

Interestingly, it was not the issue of incrementality which initially brought the left-corner algorithm to the attention of psycholinguists, rather it was the issue of computational complexity. In looking at parsing algorithms so far, we have been principally concerned with how well they model the incrementality which is clearly demonstrated by the human sentence parser. There are, however, two other criteria which are commonly taken into consideration when motivating or evaluating particular parsing algorithms, and these concern their 'computational complexity'.

- **Time:** How quickly can the algorithm find a parse tree for an input sentence?
- **Space:** How much memory does the algorithm require to build a parse tree?

We will leave discussion of time complexity until § 3.2, and for the moment concentrate on space, or memory load, considerations. We have already seen examples of garden-path sentences, where the human parser finds it difficult to obtain the correct analysis. We noted this was particularly true for the so-called reduced relative sentence in (10), or the well-known, and more pathalogical, example below:

(16) "The horse raced past the barn fell"

Many people refuse to accept this is a grammatical sentence, unless they are assisted (again, cf. *The horse that was raced past the barn fell down*). There is however another type of sentence which is equally grammatical, and difficult to interpret, but for rather different reasons. Consider first the following sentence:

(17) "The cat that the dog chased died"

This is called a *centre-embedded* sentence, since the clause about 'the dog chasing' occurs in the middle of the clause about 'the cat dying'. This sentence may seem a little awkward, but presents no real difficulty. However, grammatically speaking, there is no reason why we cannot embed yet another clause, as in:

(18) "The mouse that the cat that the dog chased bit died"

Suddenly the sentence becomes virtually uninterpretable, not because of any ambiguity in the sentence's structure, but because there is just something fundamentally difficult about constructing an interpretation for it. It has been suggested that one possible explanation for this is that processing the sentence may exceed the memory capacity of the human parser. That is, the space required by the parser to analyse such utterances exceeds the working memory available to the parser. Importantly, however, people only really have difficulty with centre-embedding, not left- or right-embedding constructions as in:

(19) "[$_S$ [$_S$ That the dog chased the cat] bothered Ted]"

(20) "[$_S$ Ted believes [$_S$ that the dog chased the cat]]"

In constructing the parsing algorithms above, we have made use of a stack as an important data structure in each algorithm. Intuitively, the stack keeps track of what categories have been predicted and therefore need to be found in the case of top-down algorithms; or it keeps track of what categories have been found, and still need to be structured together, in the case of bottom-up algorithms. It therefore seems reasonable to consider the stack size as a possible metric for syntactic memory load (though see (Frazier, 1985) for an alternative proposal), and to consider how various parsing algorithms compare with human performance. Beginning with the observation that people find left- and right-embeddings relatively easy, while centre-embeddings are difficult, Johnson-Laird (1983) observed that neither top-down nor bottom-up parsers correlate increasing stack size with centre-embeddings only. The left-corner algorithm, however, does exhibit the correct pattern of behaviour, and might therefore be considered to more accurately characterise human parsing and memory limitations.

Since this original work, there has been additional work in refining Johnson-Laird's idea (see Abney & Johnson (1991), Resnick (1992)). Stabler (1994) in particular is concerned with formulating a parsing algorithm

which both maximises incrementality, and continues to make the correct memory load predictions. The work of Gibson (1991) provides a more articulated account of precisely how memory load might be determined for syntactic structures. As we will see in §4.3, he goes on to suggest how such a memory-load account can also be used to explain a range of attachment preferences. In sum, while formulating an algorithm which accounts for all the relevant data is a difficult task, it is interesting that the simple empirical facts regarding incrementality and the difficulty of centre-embeddings can take us quite far in characterising the space of possible human parsing algorithms.

3 Ambiguity in Parsing

We observed at the beginning of this chapter that sentences may have more than one potential syntactic analysis, either *locally* at some point during parsing, or *globally* when the utterance as a whole has several possible interpretations. In the above discussion we did not discuss how this issue is to be addressed. For example, in (13, step 3.) above, the NP was expanded using the 'NP $\longrightarrow$ Det N', rule, which happened to be correct, but we could equally have chosen one of the other NP rules at that point, and then failed to reach a parse. In order to ensure that a parse is found (for grammatical sentences), some mechanism is needed to ensure that all possible parses can be considered. There are essentially three ways this can be achieved:

- **Backtracking:** Pursue a single analysis. When more than one structure can be built, choose one, but mark that decision as a 'choice point'. If that analysis cannot be completed (i.e. it reaches an incomplete state where no further rules can be applied) undo everything up to the last choice point, select a different structure, and try again. Proceed until a parse is found.

- **Determinism:** Give the parser sufficient information, such as looking ahead at the words to come, so that it can successfully decide which is the right rule to use at any given point. This means the parser won't make any 'mistakes', so no parallelism or backtracking is required.

- **Parallelism:** Pursue all possible parses in parallel, i.e. when more than one structure can be built, build all of them simultaneously, and simply discard those which don't lead to a valid parse.

Each technique above has its pros and cons when considered from the point of view of human sentence parsing. In the following sections we consider each one in turn.

3.1 Serial, Backtracking Parsers

We observed above that in (13, step 3.), the NP node was arbitrarily expanded using one particular rule instead of another (since the other would not have led to a successful parse). A serial parser, however, has no way of knowing which is the right rule to use at that particular point in the parse. It could equally well have used another rule, and subsequently failed to reach a parse. In the context of a complete language understanding model, it may be that a particular syntactic analysis, while perfectly grammatical, becomes semantically implausible as well (recall (5), and the relevant discussion). To accommodate such situations, we must augment the parser with a mechanism for *recovering* from such erroneous decisions. This is done by keeping track of the 'choice points' during the parse, i.e. those points at which there were multiple rules which could have been used to expand a particular node.

This mechanism provides us with a course of action we can take should the parse become blocked. Thus, if we reach a point where we can no longer continue with the current parse (e.g. the current word cannot be grammatically attached), we assume that a wrong decision was made at some choice point. To continue parsing, the parser selects some choice point, and restarts the parse from that point, making sure to choose a different rule to expand the node at that point. Typically, in selecting a choice point, the parser begins with the most recent. There are several reasons for this strategy:

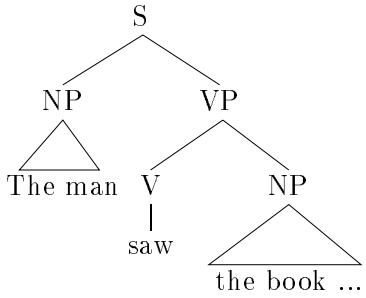
1. It seems likely that if earlier choices were indeed erroneous, they would have been discovered sooner. The fact that the parse successfully continued for some time, suggests they were good decisions. This assumes left-to-right, incremental processing.

2. All things being equal, this strategy will involve the least effort, since a smaller portion of the sentence will be reparsed.

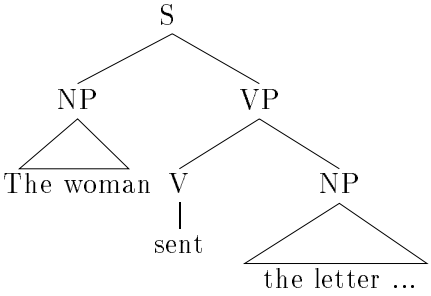
This is the strategy used in the top-down parsers described in §2.1.2, but it is not necessary for us to restrict ourselves to this approach. A superficial consideration of human parsing performance suggests, however, that it may be a good first approximation. There is substantial evidence that reanalyses

which are identified immediately (i.e. the choice point occurs just prior to the point of parse failure) are easier than cases where the appropriate choice point occurs 'further back' in the parse. Consider, for example the following two parse fragments:

(21) "The man saw the book was open"

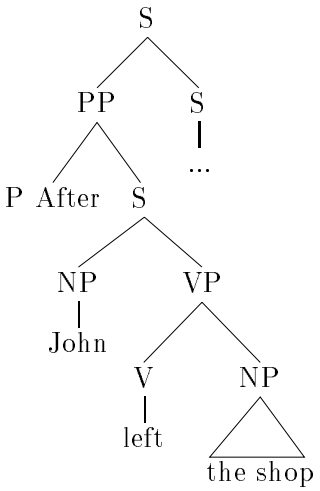


(22) "The woman sent the letter was pleased"



In (21), we need simply backtrack to the VP node, and reparse using the rule $VP \longrightarrow V \quad S$, while in (22) it is necessary to reparse from the initial subject NP, using the rule $NP \longrightarrow NP \quad RelC$, a choice point which involves substantially more reparsing. This might be accepted as an explanation for the fact that people find (21) much easier to process than (22). There are, however, counterexamples to this theory. Consider the sentence,

(23) "After John left the shop closed."



Here, it is necessary for *the shop*, initially attached at the direct object of *left*, to be reanalysed as the subject of the main clause. While this should require no more reanalysis by the parser than (21), people find it much more difficult to process. Puzzles such as this have led to parsing models which either restrict the backtracking mechanism in such a way that it makes (22) and (23) both difficult to reparse, as in the parser of Abney (1989), and models which argue that the difficulty of reparsing, or *reanalysis*, is due to independent linguistic reasons, as suggested by Pritchett (1992). We will consider these models further in §4.3.

3.2 Deterministic Parsing: Time

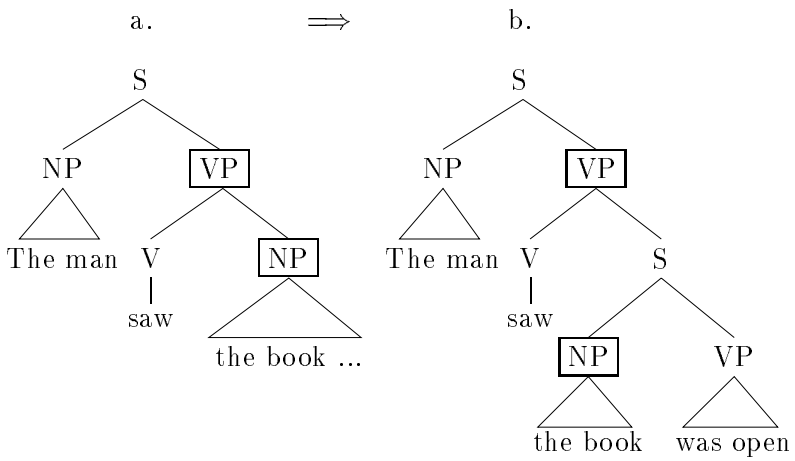
One of the most striking of our intuitions about the nature of human sentence processing is the speed at which it takes place. For most utterances, people seem able to construct an interpretation in *real time*, at least without any perceptible delay, and without any conscious effort. This observation was the prime motivation for the deterministic parser developed by Marcus (1980). The reasoning was that since the human sentence processor is fast, it must be deterministic, i.e. build syntactic analyses only when there are sufficient grounds to guarantee it is the correct one, and thus avoid backtracking. This contrasts with the *non-deterministic* 'guessing' which is the hallmark of the backtracking models discussed above. Such a deterministic parser will correspondingly *fail* if it encounters input which cannot be incorporated into the current analysis, since determinism prohibits *backtracking* to an alternative analysis. This approach has an added appeal in that it naturally predicts that failure on the part of the parser should occur for precisely those

garden-path sentences which cause humans to have conscious difficulty, e.g. (22&23), if it is to be considered psychologically plausible. To achieve such a model, Marcus implemented an LR(3) parser — essentially a bottom-up, left–to-right parser, with three item look-ahead — for English.

As we observed in §2.1.1, the ability of a bottom-up parser to leave constituents, or phrases built up so far, 'unstructured' on the stack violates the incrementality criterion established above. The extent of the look-ahead mechanism compounds the implausibility of the algorithm itself. Finally, the reliance of the parser on lexical, subcategorization information, while being quite natural for a head-initial language such as English, where verbs precede their objects, predicts serious problems for languages such as German, Dutch and Japanese, where heads may follow their arguments, rendering even three constituent look-ahead inadequate. For these cases the parser would have to leave numerous, possibly large, constituents unstructured, or 'buffered', until the subcategorizing verb was found, running counter to a variety of empirical evidence to the contrary (see Frazier (1987) for data and discussion). Another serious criticism of the model is that it makes a rather black and white distinction between those sentences which are easy to process and those which are difficult. This seems too crude, as the experimental literature has demonstrated time and again that parsing difficulty is a matter of degree, with some sentences being only marginally difficult, e.g. (21). Others, while perfectly grammatical, become almost impossible to process, e.g. (22&23). Subtle manipulation of various factors, such as plausibility and discourse, both outside the domain of the parsing explanations offered thus far, can be used to either amplify or virtually nullify the so-called garden-path effect for reduced relative sentences such as (22) (Crain & Steedman, 1985), (MacDonald, 1994) (but see also (Frazier & Clifton, 1996) and (Sturt & Crocker, 1997) for discussion).

This early work has, however, inspired a number of more psychologically plausible models which focus upon 'structural determinism' or, more accurately, structural *monotonicity*. Marcus, Hindle, & Fleck (1983) developed 'Description Theory' (or, D-theory), which suggests that parse trees be characterised in terms of a set of dominance and precedence relations. By defining trees in this way, it is possible to allow certain structural revisions which only require adding such relations, and not removing any, which is why these models are considered 'monotonic'. Recall (21), repeated below, which was an instance of easy reanalysis:

(24) "The man saw the book was open"



When the NP *the book* is parsed, we would assert that it is dominated by the VP node, and preceded by the V node. When the following VP is encountered, we must 'lower' the NP into an embedded sentence. But this is predicted to be easy, since the NP node is still dominated by the VP (both shown in boxes), and is also still preceded by the V. The only change is that the NP is no longer *immediately* dominated and preceded by the VP and V nodes, respectively. But since such a revised structure represents a monotonic increase to the parser's knowledge about the current syntactic analysis, such revision of the structure does not entail full, destructive reanalysis. A full discussion of these models would take us rather too far afield here, but for further discussion the reader is referred to the work of Weinberg (1994), Gorrell (1995), and Sturt & Crocker (1996, 1997).

3.3 Parallel Approaches

Up to this point, we have only considered parsing algorithms which pursue one syntactic analysis at a time, relying on backtracking to find different possible analyses. Such models have much to recommend them: (i) they are conceptually simpler, (ii) they are computationally simpler, in that less processing and memory resources are required, (iii) if we assume that backtracking, or *reanalysis*, correlates with increased processing complexity, then it makes strong, testable predictions about human behaviour.

There is, however, a perfectly coherent alternative which suggests that people have the ability to construct alternative syntactic analyses in parallel, when an ambiguity is encountered. That is, rather than make some decision when a choice-point is reached, simply pursue all (or perhaps some subset of) alternative parses in parallel. Thus when one particular analysis fails, it

can simply be eliminated from consideration. No backtracking is required, since we can be certain that the correct parse is taking place in parallel.

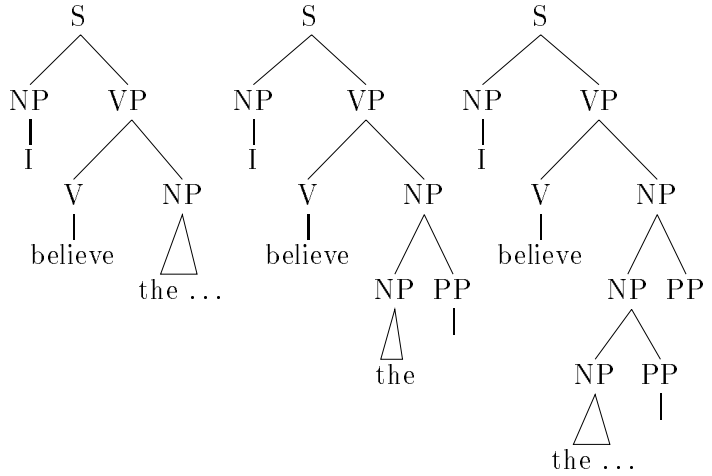
3.3.1 Bounded, Ranked Parallelism

It is important to note that full parallelism — where *every* analysis is pursued — is not psychologically possible. From a formal perspective, this is ruled out by the simple fact that there may potentially be an infinite number of such analyses, particularly if we insist on our criteria of building connected representations incrementally. Consider, for example, the following sentence initial fragment:

(25) "I believe the ..."

Each of the trees shown below is a possible partial parse for this fragment. It should also be clear that there are infinitely many other trees which are possible:

(26)



That is, for our grammar in (1), the *recursive* rule: $NP \longrightarrow NP\ PP$ means that *the* could be infinitely deeply embedded within an NP. If we accept that the mind is finite, then it simply cannot represent all the alternatives in parallel. Even if we put some arbitrary limits on the depth of the possible recursion, say ten, then there is the added ambiguity of whether or not the NP is a direct object or an embedded sentence. Contrast the following two sentences:

(27) a. "I believe [$_{NP}$ the daughter of the sister of the colonel]."
b. "I believe [$_{S}$ [$_{NP}$ the daughter of the sister of the colonel] is my aunt]."

If we combine, say, the ten analyses due to the recursion, with the two further analyses, we now have twenty. Indeed it is not difficult to introduce additional ambiguities which would multiply out the number of parallel alternatives even further. This suggests that the memory requirements of a fully parallel system would quickly exceed the short term memory resources available. Note, a further criticism of such a fully parallel system is that it would not explain the existence of garden path sentences, since in principle a parallel parser would have contructed the 'dispreferred' analysis for such sentences, thereby predicting them to be straightforward.

The solution to these criticisms has been to propose bounded, ranked parallel parsing mechanisms. By bounded we mean simply that there exists an *a priori* limit on the number of analyses we can consider in parallel. By ranked we mean that the analyses are ordered in some way. Typically, the ordering reflects the extent to which an analysis is 'preferred'. This ranking in turn accounts for the preferred interpretations exhibited by people, as people are typically aware of only one interpretation both during and after parsing.

Ranking also crucially provides the mechanism for selecting which analyses are to be pursued in parallel and which are to be discarded. That is, since ranking usually reflects some notion of preference, a bounded parallel parser will typically pursue highly ranked structures, i.e. above some rank threshold, and discard any below the threshold. This directly predicts that analyses which are discarded will be difficult garden paths, if they ultimately turn out to be correct. In contrast, there will only be some relatively small cost associated with selecting one of the other parallel (but less preferred) analyses. One such parallel model is that of Gibson (1991), which ranks parallel structures according to a set of principles based on memory load, and discards any structures which have memory requirements that are too high.

A rather more restricted parallel mechanism is the *momentary* parallelism of Altmann (1988). In this model, all possibilities are considered at each choice point, but only one 'survives' and is pursued. Altmann argued that this would permit the use of semantic and pragmatic knowledge to assist in resolving local ambiguity, while also limiting the explosion of multiple analyses that plagues full parallel models.

3.3.2 Competitive Activation

Up to this point we have considered two basic parsing mechanisms, the first in which the parser pursues a single analysis, to the exclusion of all others,

backtracking to alternatives as required. The second pursues a ranked and bounded set of parse analyses simultaneously. An additional possibility is a model which not only pursues multiple analyses in parallel, but crucially allows these structures to dynamically compete with each other in the ranking process. We might, for example, associate each competing analysis with an *activation level* where alternatives are ranked according to the strength of their activation. If we further assume some fixed total activation for the all analyses, then an increase in activation for one analysis will correspondingly entail a decrease in activation for its competitors, as in the models proposed by MacDonald, Pearlmutter, & Seidenberg (1994), Trueswell & Tanenhaus (1994), and Tanenhaus, Spivey, & Hanna (in press). In this way, one analysis might leapfrog several others in the ranking as the activation level for each analysis is adjusted during the course of parsing. Parallelism is then naturally bounded by simply dropping from consideration those analyses whose activation drops below some specified threshold.

There are numerous ways in which competitive activation might be realised in the HSPM. Stevenson (1994) proposes a hybrid parser which permits alternative syntactic attachments to compete with each other. Crucially, only a limited space of alternatives is allowed by the parser, and competition is based on syntactic information alone. In an alternative model, MacDonald, Pearlmutter, & Seidenberg (1994) argue that syntactic representations and constraints interact freely with other levels of representation. The model is therefore relatively unconstrained with regard to the space of competing structures, and also the kinds of information which are brought to bear on the competition. So semantic and discourse constraints can directly influence competition between syntactic alternatives. We expand on this issue of 'modularity' versus 'interaction' in the next section.

4 Strategies for Disambiguation

We noted at the beginning of this chapter that not only is language highly ambiguous, but that it is precisely how people cope with such ambiguity which gives us some insight into the workings of the mechanisms they employ. In the above sections we have seen how a number of different parsing architectures deal with ambiguity in general. In the following sections we consider some specific strategies which have been proposed in the psycholinguistic literature, and ask which combination of general parsing mechanism and specific strategies might best account for, and explain, existing experimental findings.

4.1 Modularity versus Interaction

In determining the strategies involved in disambiguation, a most fundamental question concerns what types of knowledge people recruit during parsing. There is no question that the process of language comprehension ultimately makes use of the vast range of linguistic and world knowledge in its efforts to arrive at an appropriate interpretation. Rather, the debate is concerned with how different knowledge sources are invoked during the time course of language comprehension, from the initial moments of perception through to the final stages of full understanding. Possible models range from a highly modular architecture— in which lexical access strictly precedes parsing, which in turn strictly precedes semantic processing, and so on — to fully interactive models which claim there is a single process which combines lexical, syntactic, semantic, and world knowledge constraints without distinction. To our knowledge, neither of these extremist positions is occupied (though for a strongly interactive connectionist proposal see (McClelland, St.John, & Taraban, 1989)), but there are proposals which clearly tend towards either end of the spectrum. We will first consider some of the general architectures which have been proposed before turning our attention towards specific theories.

4.1.1 Modular Models

Modularity draws its inspiration largely from the way linguists have carved up their theories: phonology, morphology, syntax, semantics, pragmatics, and so on have all been treated as independent phenomena. The result is that theories of each are extremely different from each other, not just in terms of the data they are concerned with, but in terms of the fundamental properties of the frameworks proposed. That is, the kinds of principles and representations which account successfully for phonological data are rather inappropriate for a treatment of semantics, while semantic formalisms are typically ill-suited for constructing grammars.[4] From a psychological perspective, it is a short (though not necessarily correct) jump to suggest that the different systems proposed by linguists might correspond to distinct systems within the human language processor.

[4]There has been recent work on developing more homogeneous sign-based linguistic theories within powerful unification-based frameworks, such as HPSG (Pollard & Sag, 1994). But despite sharing an overarching framework, the principles and representations instantiated within it still have important differences at each of the different levels of linguistic analysis.

It is also generally held that some ordering of these components is necessary. For example, semantic interpretation of the sentence (or fragments of it) can only take place after a syntactic analysis is constructed. Were this not the case there would be no need for syntax. Thus, a traditional view of the language processor is one which includes a number of distinct subsystems, which pass information in a particular direction, as illustrated in Figure 2. The question which then arises naturally is, what is the nature of the communication between the modules? Is it 'one-way'? This seems unlikely if the model is serial, since it would seem to preclude the kinds of semantically-driven backtracking which is assumed by most models. Is it unrestricted and bi-directional? This is possible, but rather difficult to distinguish from the non-modular view.

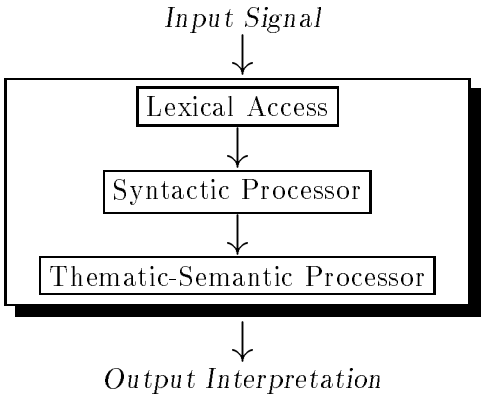


Figure 2: The Modular HSPM

Modular models have exploited the full range of parsing mechanisms from serial and parallel parsers, through to competitive activation. Perhaps the best known instance of a modular model is that of Lyn Frazier, who proposes an architecture consisting of two basic modules: a *syntactic processor* which constructs a constituent structure representation, and a *thematic processor* which selects an appropriate assignment of semantic roles for the syntactic structure, on the basis of real-world knowledge (see Frazier (1984) and Rayner, Carlson, & Frazier (1983) for discussion and data supporting this view). The distinct stages of processing provide a potential explanation for the range of relative processing effects which occur due to variations in pragmatic plausibility:

" ...it follows automatically that a sentence will be easier

> to process when the frame chosen by the thematic processor is consistent with the initial syntactic analysis of the input, than in cases where the two conflict."
>
> (Frazier, 1984)

It is assumed that the thematic processor operates concurrently with the serial syntactic processor, permitting the rejection of inappropriate analyses immediately after they are proposed. Crucially, however, Frazier maintains that the *initial* decisions concerning constituent structure are made solely by the syntactic processor without 'top-down' influence from the thematic processor. This is broadly representative of most modular models, and raises another important point. Modular models do *not* in practice assume or entail that one module must *complete* processing the utterance before it is passed to subsequent modules. Rather, they assume that each module makes its output available *incrementally* as the sentence is processed. Indeed, it seems that the principal computational advantage of a modular architecture is precisely that it enables multiple distinct, but related, processes to operate concurrently.

4.1.2 Interactive Models

An alternative position is to suggest that there is no principled, *a priori* internal structure to the human language faculty, of the sort that modularists have proposed. Rather, we might begin with the observation that our linguistic knowledge consists of a variety of heterogeneous constraints, some concerned with phonology, some with syntax, some with pragmatics, and so on. Further, since we know that many different constraints are essential to resolving ambiguity and arriving at an ultimate interpretation, the constraints should simply be permitted to interact freely during this process. Typically this view is associated with a parallel parsing mechanism where different constraints combine to eliminate or support particular analyses. Indeed, the best examples of this approach, such as the work of Tanenhaus, MacDonald and colleagues, assume a competitive architecture as discussed in §3.3.2.

To exemplify the constraint-based, interactive approach, Trueswell & Tanenhaus (1994) consider the following pair of sentence fragments:

(28) a. "The fossil examined ..."
b. "The archaeologist examined ..."

As discussed for example (10), *examined* is ambiguous with respect to its form as either a past tense or past participle. If we assume the former,

then it is attached as a main verb, and the NP is treated as the subject, and assigned the thematic role AGENT. If it is the past participle, we must construct a reduced-relative clause, and the NP will be interpreted as the object, or THEME. The semantic fit of the NP with a particular role seems to successfully disambiguate these two fragments. In (28a), *the fossil* is most likely to be the THEME (the thing that was examined), thereby promoting the reduced-relative analysis. While in (28b), *the archaeologist* is most likely to be the AGENT (the entity doing the examining), thereby promoting the simple past reading of the verb, and the corresponding construction of a simple active sentence.

The interactive position is that such 'semantic-fit' constraints will combine directly with syntactic constraints to resolve such ambiguities immediately. Two stage, modular models, on the other hand, maintain that the structurally preferred analysis (the main verb reading) will be constructed first. Then the thematic processor will accept it if the semantic fit is consistent (as in (28b)), or reject it and force the construction of a reduced-relative in examples such as (28a). The predictions of the modular versus interactive models would therefore seem to be clear. The modular model predicts that the main clause reading will systematically be preferred to the reduced-relative, while the interactive position holds that there is no such systematic preference. Unfortunately, flexibility in both models, combined with contrasting interpretations of the empirical findings, mean that neither theory has been successfully refuted. Modularists point out that the semantic fit constraints, if strong enough and early enough, will force a rapid reanalysis into the parser, such that examples such as (28a) will be quickly reanalysed to the reduced relative. This may even happen so quickly that current experimental paradigms, such as eye-movement studies, will not successfully observe the slight increase in complexity which is predicted. Furthermore, while some studies demonstrate that the garden-path effect can be eliminated, there remains a general bias towards constructing the simple active clause. The interactionists eschew this by suggesting that, though all constraints combine simultaneously, some — such as the preference to build an active over a reduced-relative clause — will have greater 'weight'.

4.1.3 Discussion

As we pointed out earlier, the real issue in the debate over the architecture of the human sentence processor, concerns *which* knowledge sources are used *when*. In its simplest form, the modular position is that constraints of a particular type (e.g. phonological, syntactic, or thematic) are

'clustered together', and have some form of temporal priority. That is, syntactic constraints operate prior to semantic ones, since if this did not occur, the semantic processor would have to consider a vast array of possible interpretations, many of which would be syntactically impossible. A further computational advantage is that each module needs only pay attention to a relatively small knowledge base, rather than the vastness of everything known about language. In response to this last point, the constraint-based view holds that the relative priority of constraints will not be determined by their 'type', but rather by their prior use and effectiveness, as determined by our linguistic experience. That is to say, the human sentence processor will give preference to constraints which are frequently applied, and which help in the interpretation process.

Interestingly, this view does not in principle exclude the modular position. Perhaps it is precisely the syntactic constraints which are most useful and most frequently applied, to such an extent that non-syntactic constraints are only brought to bear later. This would suggest that even if the system doesn't begin as a modular architecture, it may naturally develop into one. In the following sections we consider some of the leading theories regarding the strategies for sentence processing. Considered from this constraint-based perspective, each may be seen as an attempt to identify which constraints have priority in the process of interpreting language, and there validity does not necessarily hinge on specific assumptions of modularity.

4.2 Structural Strategies

Some of the most influential research on sentence processing has arisen from the work of Lyn Frazier and her colleagues. The underlying structure of the theory which has emerged assumes an organisation which is similar to that presented in §4.1.1. Frazier has concentrated on identifying the strategies which are operative at the syntactic level. Following the work of Kimball (1973), Frazier (1979) has suggested that the syntactic processor is guided by two basic principles of *Minimal Attachment* and *Late Closure*, defined as follows:

(29) **Minimal Attachment** (MA): Attach incoming material into the phrase marker being constructed using the fewest nodes consistent with the well-formedness rules of the language.

Late Closure (LC): When possible, attach incoming material into the clause or phrase currently being parsed.

This model of sentence processing is strictly incremental: lexical items are incorporated into the current partial analysis as they are encountered. Where there is an ambiguity in the direction the analysis may take, the principles of MA and LC determine the parser's decision,[5] and if they contradict each other then MA has the higher priority of the two. If the analysis chosen turns out later to be incorrect — i.e. the parser has been led down the garden-path — then the parser backtracks to pursue an alternative analysis. For this reason, Frazier's account has been dubbed the *Garden-Path Theory*. It is important to note that the notion of garden-path which Frazier adopts is very general, ranging from conscious garden-paths, which are noticeably difficult to recover from, to unconscious garden-paths which can only be observed by experimental paradigms which are sensitive to subtle but systematic increases in complexity. This differs from the simple notion of garden-path phenomena assumed by Marcus's deterministic parser — i.e. just conscious examples.

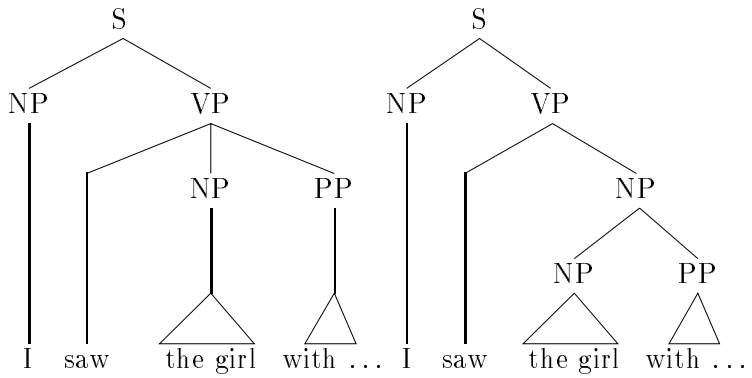
The principles of MA and LC provide a reasonable account of the core attachment preferences in ambiguous constructions. Let's reconsider the PP attachment ambiguity, which we have simplified below:

(30) Preferred VP attachment over NP adjunction.
  a. "I [$_{VP}$ saw [$_{NP}$ the girl] [$_{PP}$ with binoculars]]."
  b. "I [$_{VP}$ saw [$_{NP}$ the girl [$_{PP}$ with flu]]]."

These sentences illustrate that there are two possible attachments for the *with* PP, as either a complement of the verb, or a modifier of the NP *the girl*. Given incremental processing, the attachment of the preposition *with* must be performed before its object NP is encountered. Interestingly, this suggests that even recourse to semantic knowledge, e.g. the properties of the preposition's object, would be of no use in making this particular decision. The two possible phrase structures at this point are illustrated below:

[5] Frazier does not actually assume that MA is a fundamental strategy; rather she uses it to describe the behaviour of the human parser. Frazier assumes that there is a 'race' to build a syntactic analysis for most recent input, and that it is the simplest analysis that will be found first, thus MA should be considered a descriptive, rather than causal, strategy.

(31) a. b.



Minimal attachment dictates that the sentence processor will opt for the analysis (31a) over (31b) on the grounds that it involves postulating fewer nodes, namely the extra NP node. The prediction is that sentences where the PP continuation is consistent with the attachment into the VP (e.g. such as *binoculars*) will be easier to process. This preference has been demonstrated using eye-movement studies by Rayner, Carlson, & Frazier (1983) and replicated by Ferreira & Clifton (1986) using a similar paradigm which also tested for possible contextual effects.

If we consider the reduced relative garden-path discussed earlier (again based on data from Rayner *et al* and Ferreira and Clifton cited above), the minimal attachment analysis also applies:

(32) Preferred active clause over reduced relative.
  a. "$[_S\ [_{NP}$ The horse] $[_{VP}$ raced past the barn]] and fell."
  b. "$[_S\ [_{NP}$ The horse $[_{Rel}\ [_{VP}$ raced past the barn]]] fell]."

Here we can see that when *raced* is encountered, the two available analyses vary widely in their syntactic complexity, regardless of the particular theoretical details. In the active analysis (32a), the verb *raced* projects to a VP which is then attached to the existing root S node. The complex-NP (32b) interpretation, however, requires the adjunction to the subject NP, of a relative-clause structure (a complex structure which we have abbreviated simply as Rel in (32b)). Another example is possible local ambiguity of a complement as either NP or S:

(33) Preferred NP vs. S complement.
  a. "The scientist knew $[_S\ [_{NP}$ the solution to the problem] was trivial]."
  b. "The scientist knew $[_{NP}$ the solution to the problem]."

MA predicts the NP *the solution to the problem* will be initially analysed as the direct object (33b), since this avoids postulating the intervening S node. This prediction is borne out by the eye-movement experiment described in Frazier & Rayner (1982). That study also tested cases of clause boundary ambiguity illustrated by the following sentences:

(34) Preferred object attachment where possible.
  a. "While Mary was [$_{VP}$ mending [$_{NP}$ the sock]] [$_{S}$ it fell off her lap]."
  b. "While Mary was [$_{VP}$ mending] [$_{S}$ [$_{NP}$ the sock] fell off her lap]."

There is a strong preference for attaching *the sock* as the object of mending, as in (34a), rather than as the subject of the main clause (34b), which results in a conscious garden-path. Assuming, however, that the main clause S node is available for attachment, both analyses are equally minimal. To resolve this problem, LC prefers attachment to the VP, the most recent phrase considered by the parser, over the main S, which has yet to be analysed.[6]

4.3 Grammar-Based Strategies

Frazier's garden-path theory posits a pair of strategies[7] defined purely in terms of the *form* of syntactic structure, rather than their *content*. That is, the strategies simply count the number of nodes, or make reference to their parse tree positions, without consideration of the relations such nodes represent. Current linguistic theories, however, crucially distinguish various syntactic positions, with respect to their function and content. For example, some positions are potentially assigned case, other are reserved for constituents which can bear thematic roles, others are for modifying phrases, and so on. One might therefore imagine that not all aspects of the syntactic structure are treated equally.

In particularly influential work, Pritchett has suggested that the human sentence processor is principally concerned with satisfying the various syntactic constraints. Pritchett assumes a 'principles and parameters' style of which has emerged from the Chomsky's *government-binding* (GB) theory

[6] In most cases LC is used to explain the preferred 'low attachment' of a constituent in multiple clause sentences. Consider, for example, *I told you John bought the car yesterday*. In this sentence, *yesterday* may modify either the main or embedded clause, but there is a general preference for the latter, which is accounted for by the LC strategy, since this is the clause currently being parsed.

[7] A more detailed specification of strategies is provided in Frazier & Rayner (1988), which advances a parameterised set of principles to account for certain cross-linguistic effects. However, for present purposes, we consider only the two core strategies.

(Chomsky, 1981). In addition to rules of phrase structure, such as we have been assuming, GB theory posits a number of additional contraints on syntactic structures. In particular, it assumes that each verb has a number of thematic roles which must be satisfied, such as AGENT, PATIENT or THEME as discussed earlier. The verb *mend*, for example, has a subject which is the AGENT, and may also have an object following it, which is the THEME. The θ-criterion is simply a principle of grammar which insists that obligatory thematic roles must be assigned. Pritchett brings this principle into the parsing domain, by arguing that theta-role assignment be conducted as rapidly as possible (Pritchett, 1992):

(35) **Theta Attachment:** The θ-criterion attempts to apply at every point during parsing given the maximal θ-grid.

Roughly, this says attach constituents so as to receive the θ-roles of a given lexical item. The central assumption of this is that lexical entries of verbs are fully specified for thematic roles, as outlined above, and that such information is immediately accessed and used to drive parsing.

In addition, Pritchett suggests the following principle, to account for the cost of reanalysis in the event that the parser makes an incorrect attachment:[8]

(36) **On-Line Locality Constraint:** The target position (if any) assumed by a constituent must be governed or dominated by its source position (if any), otherwise attachment is impossible for the Human Sentence Processor.

Put simply, this states that once a constituent has been assigned a θ-role by some verb, it is difficult to move it out of that position. To make this more concrete, let's reconsider the example from (34) above, repeated for convenience:

(37) a. "While Mary was $[_{VP}$ mending $[_{NP}$ the sock]] $[_{S}$ it fell off her lap]."
b. "While Mary was $[_{VP}$ mending] $[_{S}$ $[_{NP}$ the sock] fell off her lap]."

In (37a), at the point of processing *the sock*, it is attached as an object of *mending* since this means it can successfully discharge the THEME role

[8] In the original formulation of his theory, Pritchett defined the Θ-Reanalysis constraint, which required that a constituent remain in the Θ-Domain upon reanalysis — the On-Line Locality Constraint effectively subsumes the original formulation.

for *mending*. However, should this attachment turn out to be incorrect, as in the continuation given in (37b), a garden path results and we must reanalyse *the sock* as the subject of *fell*, as shown in (37b). This reanalysis involves moving *the sock* out of the object position (and hence government domain) of *mending*, and into a new domain, i.e. of the subject position of the verb *fell*. The On-Line Locality Constraint, therefore, correctly predicts a garden path effect. In contrast, let's consider again example (33), again repeated below:

(38) a. "The scientist knew [$_S$ [$_{NP}$ the solution to the problem] was trivial]."
b. "The scientist knew [$_{NP}$ the solution to the problem]."

Pritchett argues that no garden path occurs in such sentences. As in Frazier's account, Pritchett suggests the NP *the solution to the problem* is originally licensed and attached as the direct object of *knew*, as in (38b). When the final VP is encountered in (38a), the NP is re-licensed as the subject of the embedded clause. Crucially, however, the NP remains governed by *knew*. This predicts the lack of a garden-path effect, contrary to the analysis of Frazier. This difference is due primarily to contrasting definitions of the term 'garden-path': while Frazier is concerned with characterising the broad range of processing phenomena, from subtle preferences to conscious garden-paths, Pritchett's account is exclusively concerned with the latter.

A related proposal has been made by Abney (1989), who develops a deterministic LR parser which drives attachment decisions on the basis of the thematic roles required by particular verbs. He introduces a 'steal-NP' operation to permit certain difficult, but possible, reanalysis operations as in (37), and predicts full garden paths to be unparseable. One problem shared by the models of both Pritchett and Abney derives from their reliance on verbal heads to drive the parsing process, which predicts that much of the processing in verb-final languages, such as German and Japanese, will be delayed, and therefore not incremental. To overcome this problem, Crocker (1996) proposes the following replacement for Theta Attachment:

(39) **A-Attachment** (AA): Attach incoming material, in accordance with $\overline{X}$ theory, so as to occupy (potential) A-positions.

where the simplified definition of A-position is roughly as defined in (Chomsky, 1981) (see also (Sells, 1985) for some discussion):

(40) **A-Position** Those positions (e.g. subjects and objects) which are *potentially* assigned a θ-role, e.g, subject and complement positions.

This allows us to maintain the spirit of Pritchett's Θ-Attachment strategy, while avoiding specific reference to thematic information which may not be available, particularly in verb-final languages. Specific evidence in support of this strategy (and Minimal Attachment) comes from Frazier (1987), who examines attachment preference in Dutch.

In related work, Gibson (1991) argues that unresolved thematic role assignments don't directly determine initial attachments, but rather affect the memory load associated with a particular analysis. A cost is associated with each unassigned theta role, and each NP position which has yet to receive a theta role. From this, a function is derived which ranks alternative parses: roughly, the more unresolved role assignments, the greater the load. In the context of a bounded, parallel model, syntactic analyses with lower memory load are preferred, and those which exceed a specified threshold — as occurs in some centre-embeddings and reduced-relative clauses — are dropped from consideration by the parser. This approach has the interesting property of unifying the explanation of attachment preferences, garden-path effects, and memory load effects, within a single framework.

The important hallmark of all of these approaches is that they are grounded directly in the *content* of the syntactic relations posited by current linguistic theory, rather than on the more 'artefactual' structural representations of such relations. For example, Frazier's Minimal Attachment strategy crucially relies on a rather peculiar phrase structure analysis to capture the preference in (30), above. That is, for MA to predict the attachment, Frazier has to stipulate that PP arguments are attached directly to the VP, while NP modifiers introduced an extra branching NP node. In the account presented above, this preference is accounted for directly, by claiming that theta (or argument) positions are preferred attachment sites, over modifier positions.

4.4 Experience-based Strategies

It has long been accepted that statistical effects of some sort play a role in various aspects of linguistic perception. An obvious, though not necessary, assumption is that such statistical mechanisms are derived from the frequency with which particular lexical items and structures are encountered during our linguistic experience. Several well-specified theories of frequency effects exist at the sublexical and lexical levels: Shillcock, Cairns, Chater, & Levy (to appear) suggest a statistical account of speech segmentation, Corley & Crocker (1995) argue for the role of statistical mechanisms in lexical disambiguation, Ford, Bresnan, & Kaplan (1982) for subcategorization pref-

erences, and others for the role of frequency in accessing lexical meaning (see Duffy, Morris, & Rayner (1988) and Kawamoto (1993)). More recently, theories concerning the role of statistical knowledge and strategies in syntactic processing have begun to emerge.

In most models, statistical mechanisms are taken as an *additional* factor, rather than a complete replacement, for the kinds of strategies we have discussed so far. The problem which faces researchers is: what rules, constraints, or processes are frequency-based in nature? In other words, when does the HSPM pay attention to statistical frequency effects, and when does it ignore them in favour of more general or fundamental principles? Or indeed, are there any such fundamental principles which are not simply statistical 'generalisations'? The space of possible models is vast, and at present there is not enough empirical evidence for us to reasonably select among them, but it is worth looking briefly at the perspectives which are emerging.

Within the constraint-based architectures of MacDonald, Pearlmutter, & Seidenberg (1994) and Trueswell & Tanenhaus (1994), it is envisaged that frequency effects will influence the likelihood or strength with which any particular constraint is applied, suggesting a vast number of statistical parameters. Broadly speaking, these theories argue that just as interactive, constraint-based behaviour can be naturally explained by connectionist architectures, so can the existence of frequency-based information. Indeed, within these architectures the influence of frequency is not so much strategic, as unavoidable — it is a fundamental property of the connectionist computational mechanism assumed. Precisely such frequency effects are used to explain, for example, the general bias toward the main clause versus the reduced-relative mentioned at the end of §4.1.2, since people presumably encounter more simple active sentences than reduced relatives.

An alternative is to imagine a more strategic role for statistical mechanisms, where the HSPM invokes the use of frequency in only those cases where it improves performance, i.e. leads to the correct resolution of the ambiguity. The *Tuning Hypothesis* of Mitchell & Cuetos (1991) argues that a modular syntactic processor uses frequency information exclusively for the resolution of structural ambiguities (see also Mitchell, Cuetos, Corley, & Brysbaert (1995)). Thus while a large statistical knowledge base is still required, it is substantially less than that of the interactionist models, where frequency effects are not limited to parsing. In contrast with both of these proposals, Gibson, Pearlmutter, Canseco-Gonzalez, & Hickok (1996) posit that a single statistical parameter is used to determine which of two attachment strategies has priority, to explain variation in high versus low

attachment preferences across languages.

5 Summary and Discussion

This chapter began by highlighting the problems faced by the human sentence processing mechanism in its task of constructing an interpretable representation for a given sentence or utterance. In particular, language is highly ambiguous at all levels of representation, including lexical, syntactic and semantic forms. The fact that people appear to process language incrementally, essentially on a word-by-word (and perhaps even finer grained) basis, means that we are not only faced with global ambiguity, but also with the numerous local ambiguities which occur in mid-sentence. We formalised the problem of building syntactic analyses, and characterising the 'choice-points' which handle ambiguity, by considering several parsing algorithms: top-down, bottom-up, and the combined left-corner. Broadly speaking, it is the left-corner algorithm which emerges as the most plausible model in that it is typically incremental, is highly bottom-up, and approximates certain general memory load restrictions. But it important to note that it is far from perfect; memory load phenomena appear more complex than predicted by the left-corner algorithm, and incrementality is still not fully achieved in existing algorithms (unless certain restrictions are placed on the grammars used). Also, much of current work is based on English, where verbs precede their objects, making the job of incremental parsing much easier. Greater consideration of verb-final and free word order languages is necessary, if we are to arrive at satisfying models of sentence parsing (cf. Crocker (1996), Sturt & Crocker (1996), Cuetos & Mitchell (1988), Frazier & Rayner (1988) and Frazier & Clifton (1996) as examples).

We then considered how the problem of ambiguity resolution might be dealt with by the parser, and considered deterministic, back-tracking, parallel techniques for doing so. We noted that conclusive empirical data distinguishing between serial and parallel architectures has remained elusive. Of the two, the serial model has the appeal of lower memory requirements, and a generally simpler mechanism, which is probably to be preferred in the absence of evidence to the contrary. It is also worth noting that parallel models predict that alternative, though dispreferred structures are represented by the language processor. As yet direct psychological evidence for the existence of such parallel representations has not been forthcoming.

Finally we considered a range of theories about the kind of *information* or *strategies* which might then be used to inform this process. Do people

systematically prioritise syntactic knowledge and strategies, as claimed by the modularists, or can all potentially useful linguistic (and possibly non-linguistic) knowledge contribute in a unorganised manner, as claimed by the constraint-based, interactionist camp. This polarisation of the modularity issue is probably not useful: if one begins from the assumption that language processing is simply a process of integrating a variety of constraints, then the modularist's position simply claims that syntactic constraints are 'clustered' together, and typically apply prior to semantic constraints. The extent to which this is the case remains a matter of intense study, but increasingly there is consensus that the modularity versus interaction debate is really about the *degree* of modularity (or interaction), given that the extreme modular and interactionist positions are untenable.

In the last section, we noted that the interactive, constraint-based models of Tanenhaus, MacDonald and colleagues draw much of their inspiration from work on connectionist architectures. In toy implementations, such networks have demonstrated good ability to combine different kinds of linguistic information, e.g. the system of McClelland, St.John, & Taraban (1989), and naturally exhibit frequency effects. This, combined with the fact that they are modelled (sometimes rather loosely) on human neurons, makes such mechanisms appealing as cognitive models. Closer consideration raises a number of issues, however. To date, implemented models have yet to achieve sophisticated coverage of linguistic phenomena, and it is unclear that simple, non-modular connectionist architectures will scale up successfully.

More satisfactory connectionist parsing models make increased use of explicit symbolic representations, and exhibit rather less of the micro-level frequency and interaction effects of the simpler systems (Henderson, 1994). Finally, there is increasing support from the connectionist literature for modularised network models of perception (Miikkulainen & Dyer, 1991), (Jacobs, Jordan, & Barto, 1991). In sum, it is largely the micro-level properties of connectionist systems which have promoted the constraint-based interactive approach, while we currently know very little about the macro-level properties which will undoubtably be more relevant to the necessarily large and complex models of language processing. This issue in turn bears on the question of statistical mechanisms. Simple connectionist architectures have an unavoidable frequency-based element, which may not be present at the macro-level of more complex neurally realised systems. In the latter, use of statistical knowledge may turn out to be more strategic, rather than simply automatic — a result that would be compatible with some of the alternative proposals concerning statistical mechanisms.

We have seen that the space of models of the human sentence processing

mechanism abounds with a range of proposals. Each has its own appeal, and despite major differences, each can seemingly be made to account for empirical findings relatively well. Part of this stems from the fact that many models are only partially specified and implemented, if at all. Given the complexity of the accounts proposed, implementation is increasingly essential if the true predictions of a particular model are to be made, and potentially verified or falsified. The criteria for the success of a model are many: How well does it explain human behavioural data? Is there a 'simpler' theory which can capture the same data (i.e. is the model over-complicated)? Can the model scale up to incorporate sophisticated linguistic knowledge and constraints, or is the model too naive? Is the model computationally and psychologically tractable? Does the fundamental architecture support the processing of the full range of possible human languages? Does the model make too many assumptions, or fail to account for known limitations of the HSPM?

References

Abney, S. (1989). A Computational Model of Human Parsing. *Journal of Psycholinguistic Research*, *18*(1).

Abney, S. & Johnson, M. (1991). Memory requirements and local ambiguities for parsing strategies. *Journal of Psycholinguistic Research*, *20*(3), 233–250.

Aho, A., Sethi, R., & Ullman, J. (1986). *Compilers: Principles, Techniques, and Tools*. Addison Wesley.

Altmann, G. (1988). Ambiguity, Parsing Strategies, and Computational Models. *Language and Cognitive Processes*, *3*.

Berwick, R. C. & Weinberg, A. S. (1984). *The Grammatical Basis of Linguistic Performance*. Current Studies in Linguistics. Cambridge, Massachusetts: MIT Press.

Chomsky, N. (1981). *Lectures on Government and Binding*. Dordrecht: Foris Publications.

Corley, S. & Crocker, M. W. (1995). Lexical Category Ambiguity in a Modular HSPM. Paper presented at AMLaP'95, University of Edinburgh, Edinburgh, UK.

Crain, S. & Steedman, M. (1985). On Not Being Led Up the Garden Path: The Use of Context by the Psychological Syntax Processor. In D. R. Dowty, L. Karttunen, & A. M. Zwicky (eds.), *Natural Language Parsing*, chap. 10, pp. 320–358. Cambridge, England: Cambridge University Press.

Crocker, M. W. (1996). *Computational Psycholinguistics: An Interdisciplinary Approach to the Study of Language*. Studies in Theoretical Psycholinguistics 20. Kluwer Academic Publishers.

Cuetos, F. & Mitchell, D. C. (1988). Cross-linguistic Differences in Parsing: Restrictions on the Use of the Late Closure Strategy in Spanish. *Cognition*, *30*, 73–105.

Duffy, S. A., Morris, R. K., & Rayner, K. (1988). Lexical Ambiguity and Fixation Times in Reading. *Journal of Memory and Language*, *27*, 429–446.

Ferreira, F. & Clifton, C. (1986). The Independence of Syntactic Processing. *Journal of Memory and Language*, *25*, 348–368.

Ford, M., Bresnan, J., & Kaplan, R. (1982). A Competence-Based Theory of Syntactic Closure. In J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*. Cambridge, Mass.: MIT Press.

Frazier, L. (1979). On Comprehending Sentences: Syntactic Parsing Strategies. Ph.D. thesis, University of Connecticut, Connecticut.

Frazier, L. (1984). Modularity and the Representational Hypothesis. In *Proceedings of NELS 15*, pp. 131–144, Brown University, Providence, Rhode Island.

Frazier, L. (1985). Syntactic Complexity. In D. R. Dowty, L. Karttunen, & A. M. Zwicky (eds.), *Natural Language Parsing*, chap. 4, pp. 129–189. Cambridge, England: Cambridge University Press.

Frazier, L. (1987). Syntactic Processing: Evidence from Dutch. *Natural Language and Linguistic Theory*, *5*, 519–559.

Frazier, L. & Clifton, C. (1996). *Construal*. MIT Press.

Frazier, L. & Rayner, K. (1982). Making and Correcting Errors During Sentence Comprehension: Eye Movements in the Analysis of Structurally Ambiguous Sentences. *Cognitive Psychology*, *14*.

Frazier, L. & Rayner, K. (1988). Parameterizing the Language Processing System: Left- vs. Right-branching Within and Across Languages. In J. Hawkins (ed.), *Explaining Linguistic Universals*, pp. 247–279. Oxford: Basil Blackwell.

Gibson, E. (1991). A Computational Theory of Human Linguistic Processing: Memory Limitations and Processing Breakdown. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Gibson, E., Pearlmutter, N., Canseco-Gonzalez, E., & Hickok, G. (1996). Cross-linguistic Attachment Preferences: Evidence from English and Spanish. *Cognition*, *59*, 23–59.

Gorrell, P. (1995). *Syntax and Parsing*. Cambridge, U.K.: Cambridge Universiy Press.

Henderson, J. (1994). Connectionist syntactic parsing using temporal variable binding. *Journal of Psycholinguistic Research*, *23*(5), 353–379.

Jacobs, R. A., Jordan, M. I., & Barto, A. G. (1991). Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where of Vision Tasks. *Cognitive Science*, *15*, 219–250.

Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge University Press.

Kawamoto, A. H. (1993). Nonlinear dynamics in the resolution of lexical ambiguity: A parallel distributed processing account. *Journal of Memory and Language*, *32*, 474–516.

Kimball, J. (1973). Seven Principles of Surface Structure Parsing in Natural Language. *Cognition*, *2*(1), 15–47.

MacDonald, M. C. (1994). Probabilistic Constraints and Syntactic Ambiguity Resolution. *Language and Cognitive Processes*, *9*(2), 157–202.

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994). Lexical Nature of Syntactic Ambiguity Resolution. *Psychological Review*, *101*(4), 109–134.

Marcus, M., Hindle, D., & Fleck, M. (1983). D-theory: Talking about Talking about Trees. In *Proceedings of 21st Conference of the ACL*, pp. 129–136.

Marcus, M. P. (1980). *A Theory of Syntactic Recognition for Natural Language*. MIT Press Series in Artificial Intelligence. Cambridge, Massachusetts: MIT Press.

McClelland, J., St.John, M., & Taraban, R. (1989). Sentence Processing: A Parallel Distributed Processing Approach. *Language and Cognitive Processes: Special issue on Parsing and Interpretation*, *4*(3&4), 287–335.

Miikkulainen, R. & Dyer, M. G. (1991). Natural Language Processing with Modular PDP Networks and Distributed Lexicon. *Cognitive Science*, *15*, 343–399.

Mitchell, D. C. & Cuetos, F. (1991). The origins of parsing strategies. In *Current Issues in Natural Language Processing*. Center for Cognitive Science, University of Houston, Texas.

Mitchell, D. C., Cuetos, F., Corley, M. M. B., & Brysbaert, M. (1995). Exposure-based models of human parsing: Evidence for the use of coarse-grained (non-lexical) statistical records. *Journal of Psycholinguistic Research*, *24*, 469–488.

Pollard, C. & Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. Chicago, Illinois: CSLI and University of Chicago Press.

Pritchett, B. L. (1992). *Grammatical Competence and Parsing Performance*. University of Chicago Press.

Rayner, K., Carlson, M., & Frazier, L. (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.

Resnick, P. (1992). Left-corner Parsing and Psychological Plausibility. In *14th International Conference on Computational Linguistics*, pp. 191–197, Nantes, France.

Sells, P. (1985). *Lectures on Contemporary Syntactic Theories*. CSLI Lecture Notes. Stanford, California: Center for the Study of Language and Information.

Shieber, S. & Johnson, M. (1993). Variations on Incremental Interpretation. *Journal of Psycholinguistic Research*, *22*(2).

Shillcock, R., Cairns, P., Chater, N., & Levy, J. (to appear). Statistical and Connectionist Modelling of the Development of Speech Segmentation. In Broeder & Murre (eds.), *Models of Language Learning*. Cambridge, MA: MIT Press.

Stabler, E. P. (1991). Avoid the Pedestrians Paradox. In R. C. Berwick, S. P. Abney, & C. Tenny (eds.), *Principle-Based Parsing: Computation and Psycholinguistics*, Studies in Linguistics and Philosophy, Drodrecht, The Netherlands. Kluwer Academic Publishers.

Stabler, E. P. (1994). Syntactic Preferences in Parsing for Incremental Interpretation. Unpublished manuscript, UCLA.

Stevenson, S. (1994). Competition and Recency in a Hybrid Network Model of Syntactic Disambiguation. *Journal of Psycholinguistic Research*, *23*(4).

Sturt, P. & Crocker, M. W. (1996). Monotonic syntactic processing: A cross-linguistic study of attachment and reanalysis. *Language and Cognitive Processes*, *11*(6), 449–494.

Sturt, P. & Crocker, M. W. (1997). Thematic monotonicity. *Journal of Psycholinguistic Research*, *26*(3), 297–322.

Tanenhaus, M. K., Spivey, M. J., & Hanna, J. E. (in press). Modeling Thematic and Discourse Context Effects with a Multiple Constraints Approach: Implications for the Architecture of the Language Comprehension System. In M. W. Crocker, C. Clifton, & M. Pickering (eds.), *Architectures and Mechanisms for Sentence Processing*. Cambridge, England: Cambridge University Press.

Trueswell, J. C. & Tanenhaus, M. K. (1994). Towards a Lexicalist Framework of Constraint-Based Syntactic Ambiguity Resolution. In C. C. Jr., L. Frazier, & K. Rayner (eds.), *Perspectives on Sentence Processing*, chap. 7, pp. 155–180. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Weinberg, A. (1994). Parameters in the Theory of Sentence Processing: Minimal Commitment Theory goes East. *Journal of Psycholinguistic Research*, *22*(3).