Chapter 9

Processing in Context

[After the second word of Tom wanted to ask Susan to bake a cake*] we have in the semantics a function, which we might call (*Tom want*). ... If the parser is forced to make a choice between alternative analyses, it may make reference in this choice to semantics.*
John Kimball, "Predictive Analysis and Over-the-Top Parsing"

To account for coordination, unbounded dependency, and Intonation Structure, strictly within the confines of the Constituent Condition on Rules, we have been led in parts I and II of the book to a view of Surface Structure according to which strings like *Anna married* and *thinks that Anna married* are constituents in the fullest sense of the term. As we have repeatedly observed, it follows that they must also potentially be constituents of noncoordinate sentences like *Anna married Manny* and *Harry thinks that Anna married Manny*. For moderately complex sentences there will in consequence be a large number of nonstandard alternative derivations for any given reading.

We should continue to resist the natural temptation to reject this claim out of hand on the grounds that it is at odds with much linguistic received opinion. We have already seen in earlier chapters that on many tests for constituency—for example, the list cited in (1) of chapter 2—the combinatory theory does better than most. The temptation to reject the proposal on the basis of parsing efficiency should similarly be resisted. It is true that the presence of such semantic equivalence classes of derivations engenders rather more nondeterminism in the grammar than may have previously been suspected. Although this makes writing parsers a little less straightforward than might have been expected, it should be clear that this novel form of nondeterminism really is a property of English and all other natural languages and will be encountered by any theory with the same coverage with respect to coordination and intonational phenomena. It is also worth remembering that natural grammars show no sign of any pressure to minimize nondeterminism elsewhere in the grammar. There is therefore no a priori reason to doubt the competence theory on these grounds.

The only conclusion we can draw from the profusion of grammatical nondeterminism is that the mechanism for coping with it must be very powerful.

This chapter will argue that the most important device for dealing with nondeterminism in the human processor is a process of eliminating partial analyses whose interpretation is inconsistent with knowledge of the domain under discussion and the discourse context.

The chapter will also claim that combinatory grammars are particularly well suited to the incremental, essentially word-by-word assembly of semantic interpretations, for use with this "interactive" parsing tactic. Any attempt to argue for the present theory of competence and against any other on the basis of this last observation alone would be fallacious. The methodological priority of competence arguments remains unassailable, and none of the theories currently on offer, including this one, have yet come close to descriptive adequacy as competence theories. Since all of them are compatible in principle with incremental interpretation in this sense of the term, all bets are off until the question of descriptive adequacy has been settled.

Nevertheless, we can draw the following weaker conclusion. If the program sketched in this book is ultimately successful, and CCG, together with the view of constituency and syntactic structure that it implicates, is in the end vindicated as a descriptively adequate theory of competence grammar, then it is likely that it will also be very simply and directly related to the parser as well. In other words, if it is descriptively adequate, then it is probably explanatorily adequate as well.

9.1 Anatomy of a Processor

All language-processors can be viewed as made up of three elements. The first is a grammar, which defines how constituents combine to yield other constituents. The second is an algorithm for applying the rules of the grammar to a string. The third is an oracle, or mechanism for resolving nondeterminism. The oracle decides which rule of grammar to apply at points in the analysis where the nondeterministic algorithm allows more than one rule to apply. The following sections briefly discuss these elements in turn.[1]

9.1.1 Grammar

The strong competence hypothesis as originally stated by Bresnan and Kaplan (1982) assumes that the grammar that is used by or implicit in the human sentence processor is the competence grammar itself. It is important to be clear that this is an assumption, not a logical requirement. The processors that we design ourselves (such as compilers for programming languages) quite often

do not exhibit this property. There is no logical necessity for the structures involved in processing a programming language to have anything to do with the structures that are implicated by its competence grammar—that is, the syntactic rules in the reference manual that are associated with its semantics. The compiler or interpreter can parse according to a quite different grammar, provided that there exists a computable homomorphism mapping the structures of this "covering grammar" onto the structures of the competence grammar. If the homomorphism is simple, so that the computational costs of parsing according to the covering grammar plus the costs of computing the mapping are less than the costs of parsing according to the competence grammar, then there may be a significant practical advantage in this tactic. For this reason, it is quite common for compilers and interpreters to parse according to a weakly equivalent covering grammar, mapping to the "real" grammar via a homomorphism under concatenation on a string representing the derivation under the covering grammar. For example, programming language compilers sometimes work like this, when a parsing algorithm that is desirable for reasons of efficiency demands grammars in a normal form that is not adhered to by the grammar in the reference manual (see Gray and Harrison 1972; Nijholt 1980). Such a situation also arises in artificial parsers for natural languages, when it is desired to use top-down algorithms, which can be ill suited to the left-recursive rules that commonly occur in natural grammars (see Kuno 1966 for an early example). As Berwick and Weinberg (1984, esp. 78-82) note, there is therefore no logical necessity for the structures involved in human syntactic processing to have anything to do with the structures that are implicated by the competence grammar—that is, the structures that support the semantics.

Nevertheless, similar considerations of parsimony in the theory of language evolution and language development to those invoked earlier might also lead us to expect that, as a matter of fact, a close relation is likely to hold between the competence grammar and the structures dealt with by the psychological processor, and that it will in fact incorporate the competence grammar in a modular fashion. One reason that has been frequently invoked is that language development in children is extremely fast and gives the appearance of proceeding via the piecemeal addition, substitution, and modification of individual rules and categories of competence grammar. Any addition of, or change to, a rule of competence grammar will not in general correspond to a similarly modular change in a covering grammar. Instead, the entire ensemble of competence rules will typically have to be recompiled into a new covering grammar. Even if we assume that the transformation of one grammar into

another is determined by a language-independent algorithm and can be computed each time at negligible cost, we have still sacrificed parsimony in the theory and increased the burden of explanation on the theory of evolution. In particular, it is quite unclear why the development of either of the principal components of the theory in isolation should confer any selective advantage. The competence grammar is by assumption unprocessable, and the covering grammar is by assumption uninterpretable. It looks as though they can only evolve as a unified system, together with the translation process. This is likely to be harder than to evolve a strictly competence-based system.[2]

Indeed, the first thing we would have to explain is why a covering grammar was necessary in the first place. The reference grammars of programming languages are constrained by human requirements rather than the requirements of the machines that process them. Such grammars can be ill suited to parsing with the particular algorithms that we happen to be clever enough to think of and to be able to implement on that kind of machine. It is we humans who find requirements like Greibach Normal Form tedious and who prefer grammars with left-recursive rules, forcing the use of covering grammars on some artificial processors. If convenience to the available computing machinery were the only factor determining the form of computer languages, then their grammars would take a form that would not require the use of a covering grammar at all. It is quite unclear what external force could have the effect of making natural grammars ill-matched to the *natural* sentence processor.[3]

It is important to note that the strong competence hypothesis as stated by Bresnan and Kaplan imposes no further constraint on the processor. In particular, it does not limit the structures built by the processor to fully instantiated constituents. However, the Strict Competence Hypothesis proposed in this book imposes this stronger condition.[4] The reasoning behind this strict version is again evolutionary. If in order to process sentences we need more than the grammar itself, even a perfectly general "compiler" that turns grammars into algorithms dealing in other structures, then the load on evolution is increased. Similar arguments for the need for the grammar and processor to evolve in lockstep mean that a theory that keeps such extras to the minimum is preferred.

This strict version of the strong competence hypothesis has the effect of generalizing the Constituent Condition on Rules to cover the processor. The claim is that the constituents that are recognized in the grammar (and their interpretations) will be the only structures the processor will give evidence of. Anything else we are forced to postulate is an extra assumption and will require

an independent explanation if it is not to count against the theory. Of course, such an explanation may be readily forthcoming. But if it is not, then it will remain a challenge to explanatory adequacy.

9.1.2 The Algorithm

If we believe that the natural processor must incorporate the competence grammar directly, what more must it include? According to the assumptions with which this section began, it must include a nondeterministic algorithm that will apply the rules of the grammar to accept or reject the string, together with some extra apparatus for simultaneously building a structure representing its analysis. Provided that the competence grammar is monotonic, this structure can be the semantic translation itself, rather than a strictly syntactic structure. Under this view (which has been standard in computational linguistics at least since Woods's (1970) ATN), a syntactic derivation is simply a trace of the way in which this interpretable structure was built.

The processor must also include an oracle (dealt with in section 9.1.3) to resolve the nondeterminism that the grammar allows (or at least rank the alternatives) for the algorithm. A theory will be successful to the extent that both of these components can be kept as minimal and as language-independent as possible. For this reason, we should be very careful to exclude the possibility that either the algorithm or the oracle covertly embeds rules of a grammar other than the competence grammar.

There are of course a great many algorithms for any given theory of grammar, even when we confine ourselves to the simpler alternatives. They may work top-down and depth-first through the rules of the grammar, or work bottom up from the string via the rules, or employ some mixture of the two strategies. For obvious reasons, most parsers with any claim to psychological realism work from the earliest elements of the sentence to the last, or (for the present orthography) leftmost-first, but alternatives are possible here, too.

Such algorithms require an automaton, including a working memory such as the chart mechanism discussed below, in addition to the competence grammar and a mechanism for eliminating nondeterminism. For context-free grammars the automaton is a pushdown automaton. For the classes of grammars treated in this book, it is Vijay-Shanker and Weir's (1990; 1994) generalization of the same device, the extended pushdown automaton, also discussed below. The question we must ask under the strict competence hypothesis is, how little more can we get away with? In particular, can we get away with nothing more than the theoretical minimum—that is, an algorithm that does not need to know

about anything except rules of grammar, the string, and the state of the stack and the working memory, subject to the adjudication of the oracle?

CCGs are very directly compatible with one of the simplest classes of algorithm, namely, the binary-branching bottom-up algorithms. They are most easily understood by considering in turn: (a) a nondeterministic shift-reduce parser, and (b) a chart-based deterministic left-right incremental version of the Cocke-Kasami-Younger (CKY) algorithm (Cocke and Schwartz 1970; Harrison 1978).[5]

The nondeterministic leftmost-first shift-reduce algorithm can be stated as follows:

(1) 1. Initialize the stack to the empty stack and make a pointer point to position 0 in the string, before the first word.
    2. As long as there are any words left in the string or a combinatory rule can apply to the topmost item(s) on the stack **either**:
       a. Put on the stack (shift) a category corresponding to the word that starts at the pointed-to position, **or**:
       b. Apply the combinatory rule to the topmost categories on the stack and replace them by its result (reduce).

For a simple sentence, *Thieves love watches*, this algorithm allows an analysis via the sequence shift, shift, reduce, shift, reduce (for simplicity, NPs are shown as lexically raised):

(2)

$$\frac{(S\backslash NP)/NP : love'}{S/(S\backslash NP) : \lambda p.p\ thieves'}$$

a. Shift, Shift

$$\boxed{S/NP : \lambda x.love'x\ thieves'}$$

b. Reduce

$$\frac{S\backslash(S/NP) : \lambda p.p\ watches'}{S/NP : \lambda x.love'x\ thieves'}$$

c. Shift

$$\boxed{S : love'\ watches'\ thieves'}$$

d. Reduce

One (very bad) way of making this algorithm deterministic is to choose a default strategy for resolving shift-reduce ambiguities—say, "reduce-first"—and to keep a trail of parse states including alternatives not taken, backtracking to earlier states and trying the alternatives when the analysis blocks. This will cope with the fact that all three words in the sentence are ambiguous between nouns and verbs. However, if we want to be sure that we have found not only *an* analysis, but in fact *all possible* analyses of the sentence, we must restart the process and backtrack again until all possible avenues have been examined

and no choices are left on the trail. Because naive backtracking parsers examine all possible paths in the search space, they are time-exponential in the number of words in the sentence. The source of this exponential cost lies in the algorithm's tendency to repeat identical analyses of the same substring, as when, the algorithm having mistakenly chosen the auxiliary category for the first word of the following sentence and having failed to find an analysis at the word *take*, the entire analysis of the arbitrarily complex subject NP has to be unwound and then repeated once the alternative of analyzing the first word as a main verb is taken (the example is from Marcus (1980)):

(3) Have *the students who missed the exam* take the makeup.

Because of the extra nondeterminism induced by type-raising and the associative composition rules, there is even more nondeterminism in CCG than in other grammars, so even for quite small fragments, particularly those that involve coordination, naive backtracking parsers are in practice unusable. Unlike standard grammars and parsing algorithms, because of the associativity of functional composition and the semantics of combinatory rules, CCG derivations fall into equivalence classes, with several derivations yielding identical interpretations.[6] Of course, it does not matter which member of any equivalence class we find, so long as we find some member of each. However, the search space is unacceptably large, and to ensure that we have found at least one member of all possible equivalence classes of derivation for a string of words, we are still threatened by having to search the entire space.

Karttunen (1989), Pareschi and Steedman (1987), and Pareschi (1989) discuss the use of a "chart" (Kay 1980) to reduce the search space for combinatory parsers. Chart parsing is by origin a technique for parsing CFPSG using a data structure in which all constituents that have been found so far are kept, indexed by the position of their left and right edge or boundary in the string. Each chart entry identifies the type of the constituent in question. It is common to refer to chart entries as "arcs" or "edges" and to represent the chart as a graph or network. We will be interested in the possibility that the chart may also associate other information with an arc or entry, such as the predicate-argument structure of the constituent. (For a sentence of length n, this chart can be conveniently represented as an $n \times n$ half-matrix with sets of categories as entries.)

Using a chart overcomes the main source of exponential costs in naive backtracking, arising from repeated identical analyses of a given substring. In the case of mere recognition, it is enough to make one entry in the table for a constituent of a given type spanning a given substring. For the task of finding all

distinct parses of a sentence, the chart must include an entry for each distinct analysis spanning that substring.

Since even context-free grammars can approach bracketing completeness, and do so in practice for constructions like multiple noun compounding, to similarly reduce the worst-case complexity of the parsing problem to n^3 requires the use of structure-sharing techniques to produce a "shared forest" of analyses using linked lists (see Cocke and Schwartz 1970; Earley 1970; Pratt 1975; Tomita 1987; Billot and Lang 1989; Dörre 1997).

As noted earlier, because of the associative nature of function composition, CCG parsers will potentially deliver structurally distinct derivations for a constituent of a given type and interpretation spanning a given substring—the property that is misleadingly referred to as "spurious" ambiguity.[7] If multiple equivalent analyses are entered into the chart, then they too will engender an explosion in computational costs. To the extent that CCGs approximate the bracketing completeness of the Lambek calculus version, the number of derivations will proliferate as the Catalan function of the length of the sentence—essentially exponentially.

Pareschi, following Karttunen, proposed to eliminate such redundancies via a check for constituents of the same type and interpretation, using unification. Any new constituent resulting from a reduction of existing constituents whose predicate-argument structure was identical to one already existing in the chart would not be added to it. This reduces the worst-case complexity of combinatory parsing/recognition for the context-free case to the same as that for standard context-free grammars without "spurious" ambiguity—that is, to n^3, with the same proviso that interpretation structures are shared, and with a constant overhead for the redundant reductions and for the unification-based matching entry check.[8]

Vijay-Shanker and Weir (1994) discuss the problem of generalizing the context-free algorithms to mildly context-sensitive formalisms including the present one. Because such grammars potentially introduce infinitely many nonterminal categories, generalizing the CKY algorithm discussed below to deal with them potentially makes it worst-case exponential, unless a technique of structure sharing of category entries that they describe is used. It should not be forgotten that this is strictly a worst-case complexity result. As always, caution is needed in drawing conclusions for practical average-case complexity. Komagata (1997a) suggests that average-case recognition complexity for significant practical grammar fragments for Japanese and English is roughly cubic, so that the overhead of Vijay-Shanker and Weir's technique may not be worthwhile in practical applications.

As a first step toward defining a psychologically reasonable parser, it is instructive to see a trace of an algorithm of this kind, the left-to-right breadth-first bottom-up context-free chart parser. This can be defined in terms of an algorithm that can be informally stated as follows.[9]

(4) 1. Initialize the chart to the empty chart, and make a pointer point to position 0 in the string, before the first word.
2. Until the end of the sentence is reached:
   a. Add entries corresponding to all categories of the word that starts at the pointed-to position. Make the pointer point to the next position in the sentence.
   b. As long as there is a pair of entries in the chart that can reduce, do the reduction and add an entry representing the result to the chart, unless the matching-entry test reveals that an equivalent entry is already present.

Since by definition shifting a new lexical category for the jth word can only induce new reductions to give categories whose right boundary is at position j in the sentence, an efficient way of carrying out step 2 is to ask for all i where $0 \leq i \leq (j-2)$ whether there are any such reductions. This in turn means asking for all k where $i < k < j$ whether there are entries spanning (i,k) and (k,j) that reduce. Since adding a new entry (i,j) during this process may itself enable further reductions, it is necessary to compute the new entries (i,j) bottom up—that is, by starting with $i = j-2$ and stepping down to $i = 0$. Hence, we can state the algorithm more completely and formally as follows, where *present* is the test for a matching entry already in the table, and A, B, C are category-interpretation pairs of the form $\Sigma : \Lambda$ where Σ is a syntactic category such as NP or S/NP, and Λ is a predicate-argument structure:

(5) 1. **for** $j := 1$ **to** n **do**
begin
$t(j-1,j) := \{A | A \text{ is a lexical category for } a_j\}$
2. **for** $i := j-2$ **down to** 0 **do**
begin
3. $t(i,j) := \{A | \text{there exists } k,\ i < k < j, \text{ such that } B\ C \Rightarrow A \text{ for some } B \in t(i,k),\ C \in t(k,j), \text{ and not } present(A,i,j)\}$
end
end

This algorithm is complete, in the sense that it finds all possible grammatical constituents and all complete analyses for the sentence.

. Dexter . must . know . Warren . well .

Figure 9.1
The start-state: string and empty chart

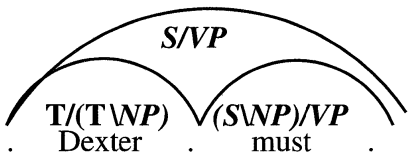

Figure 9.2
Shift *Dexter*, shift *must*, reduce

Imagine that such a parser is faced with the sentence *Dexter must know Warren well*, and suppose (simplifying) that all words have a unique category, including the adverb *well* which is a VP modifier with the single category $VP\backslash VP$. The reduce-first strategy goes through the following stages. Since the chart is initially empty, as in figure 9.1, nothing can happen until categories have been shifted to the chart for the first two words, *Dexter must*. At that point a single reduction is possible via the composition rule, leaving the chart in the state shown in figure 9.2. No further reductions are possible, so we shift a category for the next word, *know*. Two new reductions are now permitted, again via the composition rule. One of these is with the previously shifted modal, $(S\backslash NP)/VP$, and one with the result of the previous reduction, S/VP. The first induces a result that can further reduce with the subject, but this yields a result equivalent to the second in predicate-argument structural terms, so one or the other is detected to be redundant. No further reductions are possible, so the state is as in figure 9.3. We must shift a category for the word *Warren*, a shift that precipitates reductions and new entries as shown in figure 9.4. These include a number of redundant constituents that will take part in no grammatical analysis, including the *S, Dexter must know Warren*. Many of them have multiple analyses that must be detected by the matching check (or preempted by some other mechanism such as the normal form parsers that Hepple and Morrill (1989), Hendriks (1993), König (1994), and Eisner (1996) have proposed). Finally we shift a category for the adverb and halt in the state shown with a single S spanning positions 0 through 5, as in figure 9.5.

This derivation reveals the tendency for bottom-up parsers to build unnecessary constituents, here typified by the spurious *S, Dexter must know Warren*. Even the comparative simplicity of the derivation described above is mislead-

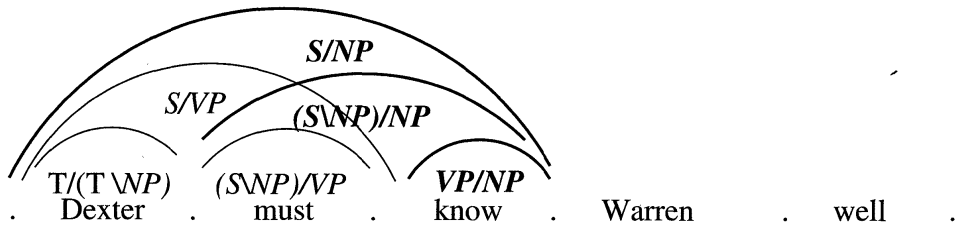

Figure 9.3
Shift *know*, reduce, reduce

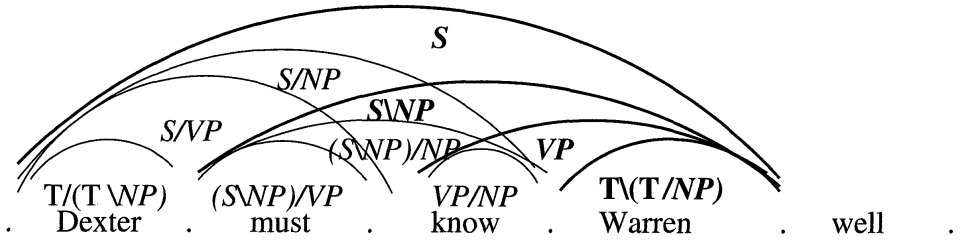

Figure 9.4
Shift *Warren*, reduce, reduce, reduce

ing in this respect. There is at least one other category for *Warren* (namely, the *subject* type-raised category), and it can combine with another category for *know* (namely, VP/S). In more complex sentences such fruitless analyses will proliferate.

However, this example serves to reify some of the main components of a practical parser, in preparation for the discussion of how this sort of device could be made more like a human processor. Human beings are rarely aware of lexical or global syntactic ambiguity in the sentences that they process, and they rarely encounter difficulty arising from nondeterminism in the grammar. How can this be? There was a broad hint in chapter 5, where we saw that intonational boundaries can on occasion reduce the ambiguity of CCG derivation. Although such indicators are frequently missing, the occasions on which they *are* missing can plausibly be argued to be exactly the occasions on which the Information Structure, and therefore some important aspects of Surface Structure, can be assumed to be known to the hearer. Perhaps there are other sources of information that mean that redundant structure building and proliferation of categories exemplified above can be eliminated for the benefit of the parser.

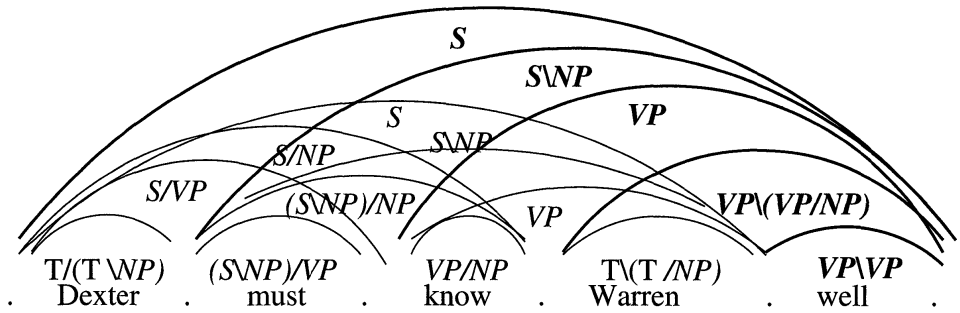

Figure 9.5
Shift *well*, reduce, reduce, halt

9.1.3 The Oracle

Nothing about the expected close relation between syntax and semantics entails that the mapping should be unambiguous, even though the grammars for artificial languages we design ourselves typically permit only the most local of nondeterminism, because they are designed for use as formal calculi. Expressions in natural languages seem to be remarkably free with ambiguity, both global and local, as in the following famous examples;

(6) Flying planes can be dangerous.

(7) a. Have the students taken the exam?
b. Have the students take the exam!

In the latter example, from Marcus 1980, the substring *have the students* is (syntactically and semantically) locally ambiguous, in the sense that a processor cannot immediately know which rule of grammar to apply after encountering it. Human beings seem to be remarkably adept at resolving such ambiguities, which are astonishingly profuse in natural language.[10]

Probably for the same reason, we do as a matter of fact tend to design our artificial languages in ways that make their symbols "locally" ambiguous, either in terms of which rule of syntax should apply, or in terms of which rule of semantics should apply. (An example of the latter is the "overloading" of an operator like + to denote distinct operations applying to integers, real numbers, and complex numbers.) The one lesson that we can derive from our experience with artificial languages and processors like compilers is that such ambiguities must be resolvable quickly and locally if the computational complexity of processing is to be contained.

In order to facilitate this requirement, programming languages are invariably carefully designed so that local ambiguity can be resolved immediately, either syntactically by examining the next symbol in the string, or semantically by examining the types of functions and arguments (as in the case of overloading above). However, natural language shows no sign of any such constraint from within grammar. For example, although the locally ambiguous substring *Have the students ...* in (7) is disambiguated by the phrase *take/taken the exam*, an indefinite amount of further linguistic material may intervene between the ambiguous substring and the disambiguating information, as when the sentences begin *Have the students who were late with the homework ..., Have the students who were late with the homework that I assigned last Monday ...*, and so on. This apparent nondeterminism in the grammar is an anomaly that requires further explanation, for if we allow the ambiguities to proliferate, then the costs of maintaining the alternatives will explode. Indeed, as Marcus points out, we must be able to eliminate all but some bounded number of alternative paths on the basis of purely local evidence, since there is no evidence that processing load increases as a worse-than-linear function of sentence length. I will call the device that eliminates nondeterminism, and decrees which rule of the grammar should be invoked at any point in the derivation, an "oracle." However this device works, it is clear that it must be very effective in order to deal with the degree of nondeterminism that natural grammars exhibit. Moreover, as noted earlier, it must also be entirely language-independent, if it is not to compromise the parsimony and modularity of the theory of the processor.

Most accounts of the human sentence-processing mechanism have assumed that local attachment ambiguity resolution is based on structural criteria, such as parsing "strategies" (Fodor, Bever and Garrett 1974; Kimball 1973), structural preferences (Frazier 1978), rule orderings (Wanner 1980), lexical preferences (Ford, Bresnan and Kaplan 1982), or lookahead (Marcus 1980). Such accounts have been claimed to explain a wide range of sentence-processing phenomena, the most spectacular of which is undoubtedly the identification by Bever (1970) of the "garden path phenomenon"—that is, the existence of sentences like the following, for which a local ambiguity is *mis*resolved in a way that makes a perfectly grammatical sentence unanalyzable:

(8) The horse raced past the barn fell.

However, such accounts have generally been characterized either by empirical shortcomings or by proliferation of devices and degrees of freedom in the theory (see e.g. the exchange between Frazier and Fodor (1978), and Wanner

(1980)). In particular, since the earliest stages of the inquiry, it has been clear that all human parsing phenomena are extremely sensitive to the influence of semantics and especially referential context. Bever (1970) notes a difference in the strength of the garden path effect in minimal pairs of sentences analogous to the following, raising the possibility of an influence either from different word transition probabilities or from the related differing pragmatic plausibility of analyzing the initial NP as a subject of the ambiguous verb/participle *sent*:

(9) a. The doctor sent for the patient arrived.
b. The flowers sent for the patient arrived.

Various computational proposals have been made for how pragmatic plausibility might have this effect via a "weak" interaction between syntax and semantics, using a filtering process of comparing rival partial analyses on the basis of their success or failure in referring to entities in the model or discourse context (see Winograd 1972 and Hirst 1987). In particular, Crain and Steedman (1985) and Altmann and Steedman (1988) proposed a criterion for selecting among analyses called the "Principle of Parsimony," which can be stated as follows:

(10) *The Principle of Parsimony*
The analysis whose interpretation carries fewest unsatisfied but accommodatable presuppositions or consistent entailments will be preferred.

These authors use the term "presupposition" in the "pragmatic" sense of Stalnaker (1974) and Lewis (1979), and explain this principle in terms of the associated notion of accommodation of unsatisfied presuppositions. They point out that the two analyses of sentence (8), which differ according to whether it begins with a simple NP *the horse* or a complex NP *the horse raced past the barn*, also differ in the number of horses whose existence in the model they presuppose (one or more than one) and in the number of properties that they assume to distinguish them—none in the case of the singleton horse, and being caused to race along a given path in contrast to some other property in the case of multiple horses. They argue that contexts which already support one or the other set of presuppositions—say, because a single horse has previously been mentioned, or several horses and some racing—will favor the related analysis at the point of ambiguity and thereby either induce or eliminate the garden path for this sentence under the Principle of Parsimony. Crucially, they also argue that the empty context, in which *no* horses and no racing have been mentioned, will favor the simplex NP analysis, because its interpretation car-

ries fewer unsatisfied but consistent presuppositions and is therefore easiest to accommodate. The principle accordingly predicts a garden path in the empty context.

In support of this view, Crain and Steedman (1985) offer experimental evidence that attachment preferences are under the control of referential context. Subjects were presented with minimal pairs of target sentences displaying local attachment ambiguities, preceded by contexts that established either two referents, respectively with and without a distinguishing property, or one referent with that property. Examples (modified from the original) are as follows:

(11) a. *Contexts:*
   i. A psychologist was counseling two women. He was worried about one of them, but not about the other.
   ii. A psychologist was counseling a man and a woman. He was worried about one of them, but not about the other.

   b. *Targets:*
   i. The psychologist told the woman that he was having trouble with *her husband.*
   ii. The psychologist told the woman that he was having trouble with *to visit him again.*

Both target sentences have a local ambiguity at the word *that*, which is resolved only when the italicized words are encountered. Frazier's (1978) Minimal Attachment Principle would predict that the second target would always cause a garden path. In fact, however, this garden path effect is eliminated when the sentence is preceded by the first context, which satisfies the presupposition of the relative-clause analysis. Moreover, a garden path effect is induced in the first target when it is preceded by the same context, because by the same token it fails to support the presupposition that there is a unique woman. Crain and Steedman (1985) also confirmed certain related predictions concerning the effect of definiteness on garden paths in the null context. The experiments were repeated and extended with improved materials by Altmann (1988) and Altmann and Steedman (1988), and the effect has been show to be robust across a number of experimental measures of processing load including brain-imaging and Event-Related Potential (ERP) measures (van Berkum, Brown and Hagoort 1999).

Whereas examples like (9) are compatible with an alternative explanation based on word transition probabilities and higher-order statistics of the language, these experiments showing effects of referential context with minimal pairs of targets are much harder to plausibly account for in this way.

The majority of early psycholinguistic experiments on processing loads used only empty contexts, and therefore failed to control for this sort of effect. However, more recent experiments (see e.g. Carroll, Tanenhaus and Bever 1978; Tanenhaus 1978; Marslen-Wilson, Tyler and Seidenberg 1978; Swinney 1979; Tanenhaus, Leiman and Seidenberg 1979; Crain 1980; Altmann 1985; Trueswell, Tanenhaus and Kello 1993; Trueswell, Tanenhaus and Garnsey 1994; Spivey-Knowlton, Trueswell and Tanenhaus 1993; Sedivy and Spivey-Knowlton 1993; and van Berkum, Brown and Hagoort 1999) have now built up a considerable body of evidence that effects of semantics, knowledge-based plausibility, and referential context are extremely strong. Indeed, almost all theories of performance nowadays admit that some such component, in the form of a "thematic processor" (Frazier 1989), "construal" (Frazier and Clifton 1996), or the equivalent, can intervene at an early stage of processing. The only remaining area of disagreement is whether anything *else* besides this potentially very powerful source of ambiguity resolution is actually required. (See the exchange between Clifton and Ferreira (1989) and Steedman and Altmann (1989)). For, if interpretations are available at every turn in sentence processing, then there is every reason to suppose that the local syntactic ambiguities that abound in natural language sentences may be resolved by taking into account the appropriateness of those interpretations to the context of utterance, even when the rival analyses are in traditional terms incomplete. Indeed, the possibility that human language-processors are able to draw on the information implicit in the context or discourse model seems to offer the only mechanism powerful enough to handle the astonishing profusion of local and global ambiguities that human languages allow and to explain the fact that human language users are so rarely aware of them. Such a selective or "weak" interaction between syntactic processing and semantic interpretation is entirely modular, as J.A. Fodor (1983, 78 and 135) points out.

If interpretation in context is the basis of local ambiguity resolution, then a number of further properties of the parser follow. The felicity of an interpretation with respect to a context is not an all-or-none property, comparable to syntactic well-formedness. Utterances are often surprising—indeed, they are infelicitous if they are *not* at least somewhat novel in content. It follows that evaluation in context can only yield information about the *relative* good fit of various alternatives. We might therefore expect the parser to use a tactic known as "beam-search," whereby at a point of local ambiguity, all alternative analyses permitted by the grammar are proposed in parallel, and their interpretations are then evaluated in parallel. Readings that fail to refer or are otherwise im-

plausible are discarded or ranked lower than ones that are consistent with what is known, along the lines suggested earlier (see Gibson 1996; Collins 1997, 1998; Charniak, Goldwater and Johnson 1998.) The parsing process then proceeds with the best candidate(s), all others being discarded or interrupted. (A similar tactic is widely used in automatic speech processing to eliminate the large numbers of spurious candidates that are thrown up in word recognition; see Lee 1989).

On the assumption that the number of alternative analyses that can be maintained at any one time is strictly limited, we can also assume that the process of semantic filtering occurs very soon after the alternatives are proposed. It should at least be completed before the next point of local ambiguity, for otherwise we incur the penalties of exponential growth in the number of analyses. Given the degree of nondeterminism characteristic of natural grammars, this means that the interplay of syntactic analysis and semantic adjudication must be extremely intimate and fine-grained. Since most words are ambiguous, semantic adjudication will probably be needed almost word by word.

For example, consider (9), repeated here:

(12) a. The doctor sent for the patient arrived.
b. The flowers sent for the patient arrived.

The garden path effect in (12a) is reduced in (12b), because flowers, unlike doctors, cannot send for things. The very existence of a garden path effect in (12a) suggests that this knowledge must be available early. If the processor were able to delay commitment until the end of the putative clause *the flowers sent for the patient*, then it would have got to the point of syntactic disambiguation by the main verb, and there would be no reason not to expect it to be able to recover from the garden path in (12a). It follows that to explain the lack of such an effect in (12b), we must suppose that the interpretation of an *incomplete* proposition *the flowers sent for* ... is available in advance of processing the rest of the PP, so that its lack of an extension can cause the garden path analysis to be aborted.[11]

However, the proposal to resolve nondeterminism by appeal to such interpretations immediately leads to an apparent paradox. If the processor resolves nondeterminism in midsentence, more or less word by word, on the basis of contextual appropriateness of interpretations, then those interpretations must be available in mid-sentence, also more or less word by word. However, under the rule-to-rule hypothesis and the strict competence hypothesis, only *constituents* have interpretations, and only constituents are available to the pro-

cessor. Now, there is no particular problem about constructing a grammar according to which every leftmost string is a constituent, so that processing can proceed in this incremental fashion. Any left-branching grammar provides an example. For such grammars, the assumption of a rule-to-rule compositional semantics means that, for each terminal in a left-to-right pass through the string, as soon as it is syntactically incorporated into a phrase, the interpretation of that phrase can be provided. And since the interpretation is complete, it may also be evaluated; for example, if the constituent is a noun or an NP, then its extension or referent may be found.

A right-branching context-free grammar, on the other hand, does not have this property for left-to-right processors. In the absence of some further apparatus going beyond rule-to-rule processing and rule-to-rule semantics, all comprehension must wait until the end of the string, when the first complete constituent is built and can be interpreted. Until that point any processor that adheres to the strict competence hypothesis must simply pile up constituents on the stack. It therefore seems that we should, under the strict competence hypothesis, expect the languages of the world to favor left- branching constructions, at least wherever incremental interpretation is important for purposes of resolving nondeterminism. However, the languages of the world make extravagant use of *right*-branching constructions—the crucial clause in (12), *The flowers sent for the patient*, being a case in point. The availability of an interpretation for what are in traditional terms nonconstituents (e.g. *the flowers sent* ... and/or *the flowers sent for* ...) therefore contradicts the strict competence hypothesis, if we assume the orthodox grammar.

It is therefore interesting that CCG makes such fragments as *the doctor/flowers sent for* ... available in the competence grammar, complete with an interpretation, and comparable in every way to more traditional constituents like the clause and the predicate. To the extent that the empirical evidence—for example, from comparison of the garden path effect in similar minimal pairs of sentences by Trueswell and Tanenhaus and colleagues—suggests that interpretations are available to the processor for such fragments, it follows that the present theory of grammar delivers a simpler account of the processor, without compromising the strict competence hypothesis. Derivations like (56) in chapter 6 show that this claim extends to the SOV case.[12]

It is important to be clear that this problem for traditional right-branching grammars is independent of the particular algorithms discussed in section 9.1.2. It applies to bottom-up and top-down algorithms alike, so long as they adhere to the strict competence hypothesis. Top-down algorithms have the ap-

parent advantage of being syntactically predictive, a fact whose psychological relevance is noted by Kimball (1973) and Frazier and Fodor (1978). However, neither algorithm of itself will allow an interpretation to be accessed for the leftmost substring, in advance of their being combined into a constituent. Therefore, neither algorithm unaided will allow word-by-word incremental semantic filtering as a basis for the oracle within right-branching constructions.

To say this is not of course to deny that incremental interpretation is possible for right-branching grammars if they do *not* adhere to the strict competence hypothesis in this extreme form. In fact, the requisite information is quite easy to compute from the rules of the grammar. It would not be unreasonable to postulate a language-independent mechanism using functional composition to map traditional grammar rules onto new rules defining parser-specific entities like S/NP.

For example, Pulman (1986, 212–213) proposes a bottom-up shift-reduce processor that includes a rule "Clear" that combines subjects like *the flowers* and a transitive verb *sent* (with the "summon" reading) on a stack, thus:[13]

(13)
$$\begin{array}{|l|}\hline VP/PP : \lambda x.summon'x \\ \hline S/VP : \lambda p.p\ flowers' \\ \hline \end{array} \Longrightarrow \boxed{S/PP : \lambda x.summon'x\ flowers'}$$

The rule Clear corresponds to an operation of semantic composition on categories on the parser's stack, as distinct from a grammatical rule. It therefore violates the strict competence hypothesis. If such violations are permitted, then it is clearly easy for a processor to gain access to interpretations more incrementally than the grammar would otherwise allow.

However, this argument cuts two ways. If such entities can be associated with semantic interpretations, why are they *not* grammaticalized? Under the assumption that grammar is just the reification of conceptual structure, why are these apparently useful concepts getting left out?

Of course, there is a lot that we don't know about the conceptual infrastructure of grammar. We are not yet in a position to say whether or not there is anything odd about those concepts that causes them to be left out. However, given that such conceptual objects seem to be accessible to the parser for resolving nondeterminism, it is interesting to remember at this point that categorial grammars of the kind discussed here already *do* grammaticalize the fragments in question. Since they do so by including composition as a component of competence grammar, they predict that the same operation should be available to the processor, under the strict competence hypothesis, rather than

requiring it as an extra stipulation and thereby violating that principle.

There is in fact no sense in which a parser using a right-branching grammar under strict competence can access interpretations for substrings that are not constituents. In the case of (12b), this means that the anomaly of the clausal reading of the substring *the flowers sent for the patient* cannot be detected until after the word *patient*. However, this is rather late in the day. The very next word in the sentence is the disambiguating main verb *arrived*. Since we know that there is a garden path effect in (12a), and we know that in context the grammatical reading can be comprehended, we have reason to believe that the human disambiguation point must be earlier, around the verb or the preposition. If so, then the degree of incremental interpretation permitted under the strict competence hypothesis for standard right-branching grammars for this construction is of insufficiently fine grain.

Stabler (1991), criticizing an earlier version of this proposal (Steedman 1989), has argued that the present claims are in error, and that incremental interpretation of right-branching constituents is in fact possible without violating the strong competence hypothesis in the strict sense used here.

Stabler does not in fact adhere to the strict form of the strong competence hypothesis. It is clear that he is assuming a weaker form of the competence hypothesis, although he gives no explicit definition (see Stabler 1991, 233, n.1). In particular, in his first worked example, (p. 226) he binds a variable *Subj* in the interpretation of the sentence to the interpretation of the actual subject, via Prolog-style partial execution in the rule *I1* (p. 208). This is possible only because he is using the grammar as a predictive parser. He uses this information to identify the fact that since the context includes only one predication over this subject, that must be the one that is to come, under a caricature of incremental evaluation similar to that used above.[14]

However, we have seen that this much is merely information that could legitimately have been built into the grammar itself, via type-raising. (In fact, this analogy seems to be effectively embodied in Stabler's second example using an LR parser (Aho and Johnson 1974), although the details here are less clear.) As in the example offered above, Stabler's processor has not actually handled any interpretations that correspond to nonconstituents. It is therefore more important to ask whether it adheres to the strict competence hypothesis in all other respects by entirely avoiding interpretation of nonconstituents of the *the flowers sent* variety, or whether it violates the hypothesis by covertly constructing interpreted objects that are not merely constituents according to the competence grammar, either in the form of dashed categories or in the form of partially instantiated semantic interpretations.

Curiously, since his paper is addressed to a predecessor of the present proposal, and even though he technically allows strict competence to be compromised, all of Stabler's examples take the first tactic. Thus, in his exegesis of the (right-branching) sentence *The joke is funny*, there is no sense in which there ever exists an interpretation of the nonconstituent *the joke is*, or indeed anything comparable to *The flowers sent* (see Stabler 1991, 215, and 232).[15] His parser offers no help with the question raised by (12). It is neither consistent with the strict competence hypothesis nor incrementally interpretative in the sense argued for here.

Shieber and Johnson (1993) have also argued against the same earlier version of the present proposal on a rather different ground. They freely admit (p. 29) that their proposal for incorporating a version of incremental interpretation in a more or less traditional grammar violates of the strict competence hypothesis. (The violation arises when they exploit the fact that the state of their LR parser encodes a shared forest of possible interpretable partial analyses in much the same way as Pulman's (1986) parser discussed above; see pp. 18–22.) However, they claim that within such a not strictly competence-based parser, incremental interpretation is actually simpler than strictly constituent-based interpretation. The reasoning behind this interesting claim is that, once interpretation of nonconstituents is allowed by the addition of extragrammatical apparatus, imposing strict competence on the system requires the reimposition of synchrony, via further additional mechanisms such as a clock or switch.

As in the simpler examples discussed earlier of incrementally interpreting parsers using categories and the stack as interpretable objects, the real force of this argument depends upon the extent to which the apparatus for interpreting LR states as encoding partial analyses can be given some independent motivation. All the earlier questions about why these interpretations are *not* grammaticalized also remain to be answered, especially when it is recalled that other competence phenomena (e.g. coordination, considered in part II) suggest that similar interpretations indeed behave like grammatical entities. Such questions are simply open, and they will remain so until the rival competence theories attain something closer to descriptive adequacy than any of them do today.

The resolution of the apparent paradox of incremental interpretation does not lie in Stabler's or Shieber and Johnson's parsers, but in the observation that strings like *Anna married* and *the flowers sent* are in the fullest sense constituents of competence grammar. We can retain the strict version of the strong competence hypothesis and continue to require the grammar to support incremental interpretation, if we also take on board the combinatory theory of

grammar. This theory offers a broader definition of constituency, under which more substrings, and in particular more left prefixes, are associated with interpretations as a matter of grammar. The interpretations of such nonstandard constituents can therefore be used to compare rival analyses arising from nondeterminism on the basis of their fit to the context, without violating the strict competence hypothesis.

9.2 Toward Psychologically Realistic Parsers

How could a reasonably efficient parser of this kind be built? One possibility is a very simple modification of the breadth-first incremental CKY parser sketched in section 9.1.2. The modification is that when a new constituent (i, j) is found, we not only check that it is not already present in the chart before adding it. We also check that it makes sense by evaluating it either with respect to a priori likelihood with respect to the knowledge base, as Winograd (1972) suggests for disambiguating compound NPs like *water meter cover adjustment screw* or (if it is a main-clause prefix) with respect to referential context, as in the flowers/doctor example (12).

We noted in the earlier discussion that we need only consider new arcs ending at j when the categories of the jth word a_j are shifted, and that it is necessary to compute the new entries (i, j) bottom up—that is, by starting with $i = j-2$ and stepping down to $i = 0$. For each i we ask for all k where $i < k < j$ whether there are entries spanning (i,k) and (k, j) that reduce. If so, then the result is added to the table t as $t(i, j)$ if it survives the matching check. The algorithm can be defined as follows. (Compare Harrison 1978, 433, and example (5) above—the present version differs only in assuming that categories are accompanied by interpretations and that all reductions are considered in the innermost loop.) A, B, C are category-interpretation pairs $\Sigma : \Lambda$ as before:

(14) 1. **for** $j := 1$ **to** n **do**
begin
$t(j-1, j) := \{A | A$ is a lexical category for $a_j\}$
2. **for** $i := j-2$ **down to** 0 **do**
begin
3. a. $t(i, j) := \{A |$ there exists k, $i < k < j$, such that $B\ C \Rightarrow A$ for some $B \in t(i,k)$, $C \in t(k, j)$, and not $present(A,i,j)\}$
b. $t(i, j) := rank(t(i,j))$
end
end

The function *rank* is assumed to order the constituents in $t(i,j)$ according to plausibility, either intrinsically or in terms of the state of the context or database and the Principle of Parsimony (10). For the sake of simplicity I will assume in what follows that the highest-ranked element is assigned a plausibility value of 1 and the rest are assigned a plausibility value of 0, although more realistically a range of values summing to 1 and/or a threshold for entry to the chart could be used.

To make the proposal more concrete, I will again assume a very simplified account of the discourse model related to an extensional version of the Alternative Semantics of Rooth (1985, 1992) used in chapter 5.[16] In particular, I will assume that a context is a database containing modal propositions as individuals, corresponding to the fact that it is possible for a person to send anything for a person, that it is possible for a person to summon a person, that it is possible for anything to arrive, that doctors and patients are persons, and that flowers are not. To further simplify, I will ignore the "send into raptures" sense of *send*. In the null context the discourse model might look something like the following, where the $\Diamond$ prefix means that the event to its right is possible and can be accommodated in the sense defined earlier, and where I use the logic-programming convention that variables are implicitly universally quantified:

(15) $person'x \wedge person'z \rightarrow \Diamond send'xyz$
$person'x \wedge person'y \rightarrow \Diamond summon'xy$
$\Diamond arrive'x$
$doctor'x \rightarrow person'x$
$patient'x \rightarrow person'x$
$flowers'x \rightarrow \neg person'x$

This database rather crudely represents the fact that propositions about people sending and summoning can be readily accommodated or added to the hearer's representation of the situation, but that in our simplified example no propositions with flowers as the subject of sending or summoning can be accommodated.

As far as the grammar goes, we have the usual problem of deciding whether nouns like *flowers* optionally subcategorize for modifiers like relatives and past participials or not. On the argument given in section 4.3.2 to the effect that anything out of which something can be right node raised must be an argument, examples like the following suggest that such modifiers are arguments:

(16) a. a few *men that I gave and women that I sold* flowers
b. the *flowers sent for and chocolates given to* the patient

Continuing to simplify, I will represent this by simple lexical ambiguity on the noun *flowers*. For the same reason I will also continue to assume that type-raising applies lexically.

Consider what happens when the sentence *The flowers sent for the patient arrived* is processed in this context. We shift the definite article *the* and the two categories for the noun *flowers*, which can then reduce to yield a subject (among other irrelevant raised categories) meaning something like the following, in which ι is Russell's definite existential quantifier (Russell 1905—see van der Sandt 1988; Beaver 1997):

(17) a. $S/(S\backslash NP) : \lambda p.\iota x.flowers'x \wedge px$
b. $(S/(S\backslash NP))/(N\backslash N) : \lambda q.\lambda p.\iota x.(flowers'x \wedge qx) \wedge px$

The ι operator in the first category requires the existence of exactly one entity of type *flowers'* The ι operator in the interpretation of the second category requires the existence of exactly one entity of type *flowers'* having one other property q

There are no such entities in the database, but they can be consistently accommodated. Since the first category requires accommodating one proposition and the second requires accommodating two, the first is more plausible. It is therefore ranked 1, and the accommodation is carried out. The second is ranked 0 and not accommodated. Importantly, both categories remain in the table.

Let us represent the accommodation by existentially instantiating the flowers with an arbitrary constant—say, $gensym'_1$—and adding the following fact to the database:

(18) $flowers'gensym'_1$

We next encounter the word *sent*, which has three categories:

(19) a. $((S\backslash NP)/PP)/NP : \lambda x.\lambda y.\lambda z.send'yxz$
b. $(S\backslash NP)/PP : \lambda x.\lambda y.summon'xy$
c. $(N\backslash N)/PP : \lambda x.\lambda p.\lambda y.p\ y \wedge send'yxsomeone'$

The raised subject (17a) can compose with the categories (19a,b) to yield the following categories:

(20) a. $S/PP : \lambda y.\iota x.flowers'x \wedge summon'yx$
b. $(S/PP)/NP : \lambda y.\lambda z.\iota x.flowers'x \wedge send'zyx$

The other subject category, (17b), can compose with the last verbal category, (19c), to yield the following category:

(21) $(S/(S\backslash NP))/PP : \lambda y.\lambda p.\iota x.(\mathit{flowers}'x \wedge \mathit{send}'yx\mathit{someone}') \wedge px$

To assess the plausibility of (20a,b), the processor must ask if it is possible for flowers to send things for people or send for people:

(22) a. $\Diamond \mathit{flowers}'x \wedge \mathit{summon}'yx$
b. $\Diamond \mathit{flowers}'x \wedge \mathit{send}'zyx$

Neither possibility is supported by the database, so both of these categories are associated with a low probability by the ranking function *rank* when entered in the chart.

The plausibility of category (21) depends on there being just one thing around of type *flowers'* with the property that they were sent. Although there is no corresponding proposition in the database, the knowledge base does at least support the possibility of sending flowers:

(23) $\Diamond \mathit{flowers}'y \wedge \mathit{send}'zyx$

Category (21) therefore ends up as the highest-ranked category for *The flowers sent* . . . , ranked 1 on entry to the chart. Its presuppositions are accommodated by adding the following facts to the database about the already present arbitrary flowers gensym_1:

(24) $\mathit{send}'z\ \mathit{gensym}'_1\mathit{someone}'$

The next word to shift is the preposition *for*, which first reduces with (20b) and (21) by composition. Since flowers cannot send for anything, and can be sent for people, the first is ranked 0 and the second 1 on entry to the chart:

(25) a. $S/NP : \lambda y.\iota x.\mathit{flowers}'x \wedge \mathit{summon}'yx$
b. $(S/(S\backslash NP))/NP : \lambda y.\lambda p.\iota x.(\mathit{flowers}'x \wedge \mathit{send}'yx\mathit{someone}') \wedge px$

Note that this preference for the modified subject reverses that on the subject alone. (Other reductions, which we will come back to later, are possible at this stage.)

Next we shift *the*, shift *patient*, and reduce to yield a number of categories as in the case of the subject, of which the following (where $NP^{\uparrow}$ schematizes as usual over various raised categories) carries fewest presuppositions/entailments and is highest ranked:

(26) $NP^{\uparrow} : \lambda p.\iota x.\mathit{patient}'x \wedge px$

A unique patient must therefore be accommodated using another arbitrary constant:

(27) $patient' gensym'_3$

The raised NP can combine with both categories in (25) for *The flowers sent for* .... Since patients are people and can be summoned and sent things, two categories go in the chart for *The flowers sent for the patient* ... :

(28) a. $S : \iota y.patient' y \wedge \iota x.flowers' x \wedge summon' yx$
b. $S/(S\backslash NP) : \lambda p.\iota y.patient' y \wedge \iota x.(flowers' x \wedge send' xysomeone') \wedge px$

The first of these is again implausible. The second is plausible to the extent that it is possible to send flowers for a patient, and that there is exactly one thing with the property *flowers'* and one with the property *patient'*, and that a proposition subsuming the following one—namely, (24)—is already accommodated:

(29) $send' gensym'_2 gensym'_1 someone'$

The subject in turn can combine with the main verb *arrived* and complete the analysis, since anything can arrive.

This version of the CKY parser is complete, in the sense that it builds all legal constituents, even when they are zero-ranked for likelihood. Other constituents will be built, some of which will be rejected under the matching-entry test as being redundant without any further evaluation and without affecting the rankings already assigned to the equivalent constituents already in the chart. The important thing to note is that the anomaly of the tensed-verb reading is apparent as soon as the ambiguous word *sent* is encountered.

The analysis of the sentence *The doctor sent for the patient arrived* is identical, except that because of the plausibility of doctors sending for people, and the lesser presuppositional demand of the simple NP, the tensed-verb analysis is favored over the modifier analysis until the disambiguation point:

(30) $S/(S\backslash NP) : \lambda p.\iota y.doctor' y \wedge \iota x.(patient' x \wedge send' xysomeone') \wedge px$
$S : \iota y.doctor' y \wedge \iota x.patient' x \wedge \wedge summon' xy$

Now suppose that a single doctor is already identified in the context.

(31) $doctor' dexter'$

Here the analysis of *The doctor sent for the patient arrived* will proceed exactly as in the null context, except that the unique doctor will no longer need to be accommodated. The analysis will receive a low rank for the same reason.

However, consider the case where there are two known doctors in the context:

(32) *doctor′dexter′*
doctor′warren′

Even if the entailment of the restrictor (that someone sent one of them for the patient) is not known and must be accommodated, the simple NP will fail to refer from the start and the complex NP will be highly ranked, as in the case of *The flowers sent for the patient arrived.*

We have already noted that the CKY algorithm as described here is complete. This means that it does not of itself predict exactly which sentence-context pairs will lead to unrecoverable garden paths. Moreover, we have unrealistically assumed that the ranking function does not need to take into account the preference values of the inputs to combinations. However, the algorithm shows that alternatives can be ranked consistently with the observed effects up to the point of disambiguation. This means that less conservative algorithms such as the beam-searching CCG parser of Niv (1993, 1994), the best-first chart-parser of Thompson (1990), or a version of CKY in which continuous probabilities are used to calculate exact likelihood values—say, using methods discussed by Collins (1996, 1997, 1998)—and subjected to a threshold, can be used to make more precise predictions.

9.3 CCG Parsing for Practical Applications

As noted earlier, the CKY algorithm has worst-case time complexity n^3 for recognition in the context-free case (because it involves three nested loops of complexity order n), and the n^6 worst-case complexity of recognition for Vijay-Shanker and Weir's (1990; 1993; 1994) generalization to CCG depends upon a complex structure-sharing technique for categories. Moreover, polynomial worst-case complexity for the corresponding parsers depends in both cases upon similarly subtle techniques for structure sharing among parse trees or interpretable structures using devices like "shared forests" (Billot and Lang 1989).

However, experiments by Komagata (1997a, 1999) with a CKY parser for hand-built CCG grammar fragments of English and Japanese for real texts from a constrained medical domain suggests that average-case parsing complexity for practical CCG-based grammars can in practice be quite reasonable even in the absence of semantic disambiguation or statistically-based optimization, despite its worst-case exponentiality.[17] To the extent that the psychological processor limits attachment ambiguities by the kind of semantic strategy outlined here, or by the related probabilistic techniques discussed in

section 9.2, psychologically implausible mechanisms to manage shared forests as a representation of the alternative parsers may also be eliminated.

The modified CKY algorithm is only one among a number of possibilities, including parsers based on more predictive algorithms such top-down and mixed top-down and bottom-up algorithms, such as that of Earley (1970). The important point in terms of psychological realism is that by using CCG as the grammatical module, all such algorithms can be semantically incremental, while remaining entirely neutral with respect to the particular theory of grammar involved and exactly as incremental as the grammar itself allows, in keeping with the strict competence hypothesis.

Nevertheless, for many applications this kind of algorithm will always be vulnerable to well-known limitations on our ability to represent our everyday knowledge about practical domains in ways that will support adequate assessment of plausibility. For many applications such a parser will therefore benefit very little from the pruning step. Moreover, as we saw at the end of chapter 8, this particular algorithm, being bottom-up, inherits the disease of building useless constituents.

Many of the worst effects of the disease can be eliminated by being more careful about which categories for a_j are input to the algorithm in step 1. For example, almost all nouns like *thieves* in English can be either N or NP. However, when preceded by an article, as in *the thieves*, the relevant category is, with overwhelmingly high probability, N rather than NP. This fact provides the basis for a number of low-level, purely stochastic, syntax-independent "part-of-speech-tagging" (POS) methods for disambiguating lexical form-class (Jelinek 1976; Merialdo 1994), based on algorithms that can be automatically trained on text or speech corpora. POS tagging can be used to limit the candidate categories input to the CKY algorithm to the "n-best" or most likely categories, as de Marcken (1990) points out, eliminating much of the disadvantage of bottom-up techniques.

Indeed, it is likely that CCG and other lexicalized grammars, such as TAG, will benefit more from such stochastic filtering. POS tagging is commonly based on around 60 form classes (Francis and Kučera 1964, 23–25), some of which, such as VBZ (verb, 3rd person singular present), are not as informative as they might be. By contrast, CCG and TAG have several distinct categories or elementary trees corresponding to VBZ, distinguishing intransitive and a number of distinct varieties of transitive and ditransitive verbs. This suggests that better POS-tagging algorithms could be developed by using CCG or TAG categories in place of the standard POS categories, a proposal

that has been investigated by B. Srinivas and Joshi (1994). Experiments of this kind are reported by B. Srinivas (1997) and Doran and B. Srinivas (to appear), with promising results. One interesting question for this research is whether stochastic methods will be effective in disambiguating type-raising.

Recent work by Collins (1996, 1997, 1998) points to the advantages of an even greater integration of probabilistic information with syntax in parsers for lexicalized context-free grammars, including CCG. Collins (1997, 1998) presents a technique for supervised learning of probabilistic Dependency Grammars, in which the production rules are induced from a tree-bank and probabilities are found for a given rule applying with a given lexical "head" and arguments with other given heads. Some of the distinctive features of the procedure for assigning these probabilities are a method based on maximum likelihood estimation and a "backing off" method for use in the face of sparse data. The probabilities can be used to guide search in any one of a number of standard parsing algorithms, including shift-reduce, beam search, and CKY. This parser was at the time of writing the most accurate wide coverage parser by the standard measures, with precision/recall figures better than 88% on unseen Wall Street Journal text.

Because of its simultaneous clean theoretical separation between competence grammar, parsing algorithm, and probability, and because of its close coupling of these elements in processing, Collins's method is extremely general. More or less any class of lexicalized grammar can be made to yield dependency structures, and so probabilistic parsers can in principle be induced for them too, provided that the tree-bank that is used records the relevant dependencies. Combinatory Categorial Grammar is a particularly interesting case to consider, not only because of the historically close relation between Categorial Grammar and Dependency Grammar, and because the Logical Forms that CCGs build capture the dependencies in question, but also because of the simple way in which they project lexical dependencies, and hence the associated head-dependency probabilities, onto unbounded and fragmentary constructions including coordination. (The advantages of this property for grammar induction have already been mentioned in connection with human language acquisition.)

It is not clear what psychological reality such stochastic methods can lay claim to. It does not seem likely that the semantic methods I have advocated for resolving attachment ambiguities will solve the problem of lexical ambiguity. On the other hand, it does seem possible that human processors could derive plausibility measures directly from properties of the syntax and associa-

tive properties of the memory for concepts underlying word meanings, rather than by collecting higher-order statistics over large volumes of data. The rule-based POS taggers of Brill (1992) Voutilainen (1995), Kempe and Karttunen (1996), and Cussens (1997), and related sense-disambiguation work by Resnik (1992) using WordNet (Miller and Fellbaum 1991) are suggestive in this respect. However, to the extent that very transient changes to the sets of referents that are available in the discourse model can affect processing load and garden path effects, as in the experiments discussed earlier, processes of active interpretation, including limited amounts of inference, seem to be the only plausible basis for resolution of structural or attachment ambiguities by the psychological processor.

Chapter 10
The Syntactic Interface

Lofty designs must close in like effects.
Robert Browning, "A Grammarian's Funeral"

This book began by stating some uncontroversial assumptions in the form of the rule-to-rule condition and the competence hypothesis, deducing the even more widely accepted Constituent Condition on rules of competence grammar. The Introduction also endorsed the methodological priority of investigating competence syntax over performance mechanisms. Having noted the difficulties presented by coordination and intonation in relation to the Constituent Condition on Rules, part I of the book went on to advance an alternative combinatory view of competence grammar under which these apparently paradoxical constructions were seen to conform to that condition after all. After putting the theory through its syntactic paces in part II, the progression has been brought full circle in part III by deriving some consequences for the theory of performance under a "strict" version of the competence hypothesis.

10.1 Competence

The competence theory that was developed along the way is conveniently viewed in terms of a third and final version of the by-now familiar Y-diagram in figure 10.1, which combines figures 4.1 and 5.3, again including mnemonic exemplars of the constructs characteristic of each module of the theory. According to this theory, lexical items and derived constituents (including sentences) pair a phonological representation with a syntactic category (identifying type and directionality only) and an interpretation. Chapter 5 showed that the interpretations of the principal constituents of the sentence correspond to the information structural components called theme and rheme. These in turn combine by function application or "β-normalization" to yield fairly standard quantified predicate-argument structures or Logical Forms. Predicate-argument structures preserve fairly traditional relations of dominance and command. In par-