

# Monotonic Syntactic Processing: A Cross-linguistic Study of Attachment and Reanalysis

Patrick Sturt and Matthew W. Crocker

*Centre for Cognitive Science, University of Edinburgh, Edinburgh, UK*

This paper describes a new addition to the family of parsing models based on the principles of Description Theory (Marcus, Hindle, & Fleck, 1983). We demonstrate how the definition of two simple parsing operations, *simple attachment* and *tree lowering*, which are related to the grammatical composition operations of *substitution* and *adjunction* in the Tree Adjoining Grammar formalism, yields a parser which is more constrained than previous Description Theory based models (e.g. Gorrell, 1995a). Since the tree-lowering operation allows the parser to reanalyse in the case of “unconscious” garden paths, it can be used to investigate the consequences of adopting various search strategies for reanalysis, predicting preferences in cases where more than one possibility for reanalysis exists. Considering data from English and Japanese, we show that reanalysis preferences may differ between head-final and head-initial languages, and suggest some reasons why this might be so.

## INTRODUCTION

A psychologically plausible model of human syntactic processing must provide a principled account of the extent to which the parser can “change its mind” about previously made choices during the course of a parse. For example, while experimental evidence shows us that people initially adopt the globally incorrect reading of the following locally ambiguous sentences, it is generally accepted that, on encountering the disambiguating word

---

Requests for reprints should be addressed to Patrick Sturt, Centre for Cognitive Science, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, UK. E-mail: sturt@cogsci.ed.ac.uk.

Some of the material reported in this paper has been presented in talks given to the 1995 CUNY Sentence Processing Conference in Tucson, AZ, and the 1995 EACL Conference in Dublin, Ireland. An earlier version of the paper appears in *Edinburgh Working Papers in Cognitive Science*, Vol. 11, *Incremental Interpretation*, edited by D. Milward and P. Sturt (1995). We would like to thank Holly Branigan, Chuck Clifton, David Milward, Martin Pickering and an anonymous reviewer for extremely helpful comments on earlier drafts of the manuscript.

Patrick Sturt was supported by ESRC Studentship R00429334338.

(shown here in bold type), reanalysis is more costly in (1b) and (1c), where it causes noticeable difficulty, than in (1a), where it does not:

- 1a. John knows the truth **hurts**.
- b. While Philip was washing the dishes **crashed** onto the floor.
- c. The boat floated down the river **sank**.

In building models to account for differences such as these, researchers have sought to constrain the space of possible structural revisions available to the parser, so as to include the revisions necessary for *unconscious* reanalysis (as in 1a), but to exclude those necessary for *conscious* reanalysis (as in 1b and 1c). There have been various proposals for constraining this space. One approach is to define the structural relations which may hold between the source and target positions of a constituent whose configurational position is to be altered on reanalysis (Pritchett, 1992). A second approach is to allow the parser to use certain limited destructive operations, such as Abney's (1987, 1989) *steal*,<sup>1</sup> or Lewis's (1993) *snip* operator.

A third approach is to find a representational formalism which allows just enough underspecification to accommodate the required class of structural revisions. One such representational system which has recently been employed successfully by psycholinguists is that provided by Description Theory (henceforth "D-theory") (Marcus, Hindle, & Fleck, 1983), and it is this approach which we discuss in detail in this paper.

## D-theory and Psychological Models

D-theory allows a parser to use tree descriptions, stated in terms of dominance (as opposed to *immediate* dominance) between nodes. Since dominance is a transitive relation, this means that if node *X* is asserted to dominate node *Y* at some particular parse state, then an unlimited number of further nodes may subsequently be inserted into the domination path between *X* and *Y* without falsifying the original domination statement. This property of D-theory has been exploited in some recent psycholinguistic models. For example, Weinberg (1993, 1995) and Gorrell (1995a) predict that unconscious syntactic reanalysis will be possible if the required revision can be made without falsifying any structural relation asserted in a previous state.

From a theoretical point of view, the attraction of these models is that they offer the possibility of deriving the required constraints on structural revisions "for free", as a consequence of the representational formalism used by the parser, removing the need to stipulate the constraints separately. By the same token, however, these models suffer from a conceptual

---

<sup>1</sup>Abney's "steal" operation is designed to allow for what he calls "weak" garden paths which include (1b) but exclude totally unrecoverable examples such as (1c).

problem, which becomes particularly clear when one attempts to provide a computational implementation. The problem can be summarised as follows. The most natural explanation for the constraints on the class of possible structural revisions observed in the human parser is that they are a consequence of a search strategy which is streamlined to allow for incremental, efficient processing. However, the mere adoption of the D-theory representational formalism in no way guarantees that search will be limited in the required manner. In particular, if no explicit search strategy is given, then the D-theory formalism assumes the status of a filter, or “clash check”, on possible parsing actions proposed by a “generate-and-test” processing algorithm. Such a generate-and-test algorithm would represent a totally unconstrained and possibly non-terminating search, seemingly at odds with the idea that the constraints on structural revisions are a result of a limited search space available to a streamlined processing device.<sup>2</sup> On the other hand, if it is possible to limit the choices of the parser to a constrained set of parsing operations, which are guaranteed to preserve D-theory coherence, then we can derive the required constraints on reanalysis, while simultaneously pruning the search space, thus addressing the efficiency question also. Such an approach is also desirable on a methodological level, since the explicit statement of restricted parsing operations and search strategies for their application is likely to result in a more restrained, and thus more predictive, model.

In this paper, we describe a model which builds representations that preserve the D-theory coherence conditions, but where this constraint is obtained as the result of restricting the operations available to the parser, rather than imposing a filter on proposed output. In particular, we propose that the parser proceeds by incrementally assembling descriptions of lexically anchored tree fragments, and that it has at its disposal two composition operations, *simple attachment*, which identifies the root of one tree with a node on the fringe of another, and *tree lowering*, which inserts one tree inside another at an intermediate point on the right frontier. These operations are related to the operations of *substitution* and *adjunction* found in tree-adjointing grammars (henceforth TAGs) (Joshi, Levy, & Takahashi, 1975; Schabes, Abeillé, & Joshi, 1988), though we will see that an important difference between TAG adjunction and the tree-lowering operation means that the latter, but not the former, can be used for reanalysis.

There has been a discernible trend towards constraining the power of the original D-theory model in order to sharpen its psychological plausibility. For example, Weinberg (1993, 1995) proposes a model which does not incorporate a look-ahead buffer, thus enforcing incremental, word-by-word processing. Gorrell (1995a,b) goes further still, adding precedence relations between non-terminal nodes, and insisting on full specification of node

---

<sup>2</sup>Though the clash-check itself need not be too inefficient (Cornell, 1994).

labels where Weinberg allows underspecification. In the implementation described in this paper, we take the constraints of Gorrell's model as a starting point, but, by limiting the operations of the parser, we derive a parser which is more restrictive than Gorrell's, in the sense that only a subset of attachments licensed in Gorrell's model is licensed in the model proposed here. We believe that this course of action has the following advantages over previous D-theory-based models:

1. By limiting the attachment operations available to the parser, we directly address the search problem; difficult reanalysis is unavailable simply because it is not in the parser's repertoire, and not because it is filtered out as part of a generate-and-test procedure. This has advantages for the computational efficiency of the parser, but also yields a model which is more restrictive, in the sense that only a subset of the D-theory valid attachments/reanalyses are possible. This gives the model greater predictive power.

2. The explicit definition of a set of parsing operations, which are related to a well-understood grammar-formalism (TAG), makes the model relatively simple to implement computationally.

3. Finally, psycholinguists have recently begun to give serious consideration to the question of *how* the parser reanalyses (Fodor & Inoue, 1994), as opposed to *what structures* the parser can and cannot reanalyse. This implies a need for computational models capable of making explicit predictions on this question. The formulation of parsing operations, and search strategies for their application, as we provide in this paper, represents a contribution to such a research programme.

## CONSTRAINTS ON GORRELL'S PARSER

As mentioned in the previous section, we have taken the constraints of Gorrell's model as a starting point for the parser proposed here. This model was chosen because it represents a particularly constrained, and therefore a particularly interesting, version of the D-theory hypothesis. In this section, we summarise the constraints of Gorrell's original model and discuss some of their consequences. In the following and subsequent sections, we go on to give a detailed description of our own implemented model.

### Informational Monotonicity and Structural Relations

Unlike Weinberg (1993), who allows the parser to update the set of relations by fleshing out underspecified node labels in the description carried over from the previous state, Gorrell does not permit underspecification of node labels. The parser is monotonic in the sense that all it can do to the set of structural relations is add more relations, though, in this case, the domain of representation to which this constraint may apply is limited to the purely

configurational notion of *structural relations* (i.e. dominance and precedence relations), while licensing relations (such as theta and case assignment) are not so constrained. The claim of these models, then, is that the core parser will be able to perform reanalysis just in those cases where such an action does not violate informational monotonicity.

### Conditions on Trees

We will say that the set of relations describing the phrase marker being constructed is coherent if it conforms to the following conditions for trees (adapted from Partee, ter Meulen, & Wall, 1993):

1. *Single root condition*: there is a single node, the *root node*, which dominates every node in the tree:

$$\exists x \forall y \cdot \text{dom}(x,y)$$

2. *Exclusivity condition*: no two nodes can stand in both a dominance and a precedence relation:

$$\forall x,y \cdot \text{prec}(x,y) \vee \text{prec}(y,x) \leftrightarrow \neg \text{dom}(x,y) \wedge \neg \text{dom}(y,x)$$

3. *Inheritance* (a.k.a. the “non-tangling” condition): all nodes inherit the precedence properties of their ancestors:

$$\forall w,x,y,z \cdot \text{prec}(x,y) \wedge \text{dom}(x,w) \wedge \text{dom}(y,z) \rightarrow \text{prec}(w,z)$$

Dominance and precedence are both defined as transitive relations. In addition, it is usually assumed that dominance is reflexive (every node dominates itself) and precedence is irreflexive. That is, dominance defines a *weak* partial order and precedence defines a *strict* partial order.

### Primary and Secondary Relations

Gorrell divides syntactic representation into *primary* relations (dominance and precedence) and *secondary* relations (theta-role assignment, c-command, case-assignment, etc.), of which only the primary relations are constrained by monotonicity.<sup>3</sup> This is what he calls “structural determinism”. Recall (1) repeated below in (2):

- 2a. John knows the truth hurts.
- b. While Philip was washing the dishes crashed onto the floor.

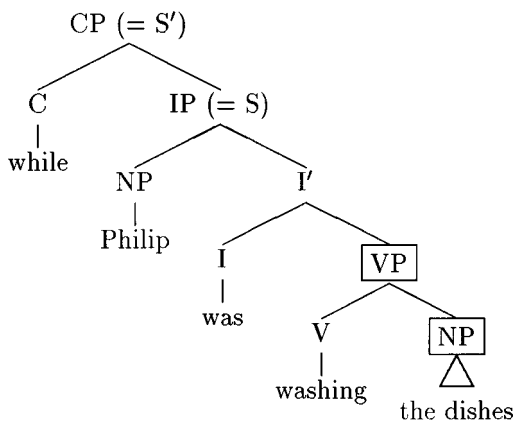
Gorrell explains the difficulty of processing (2b) in terms of the fact that monotonicity is not preserved in primary relations. In (2a), by contrast, although secondary relations have to be altered during the parse, primary relations can be built up monotonically, and the sentence is therefore predicted to be processable by the core parser. Below we give a brief

---

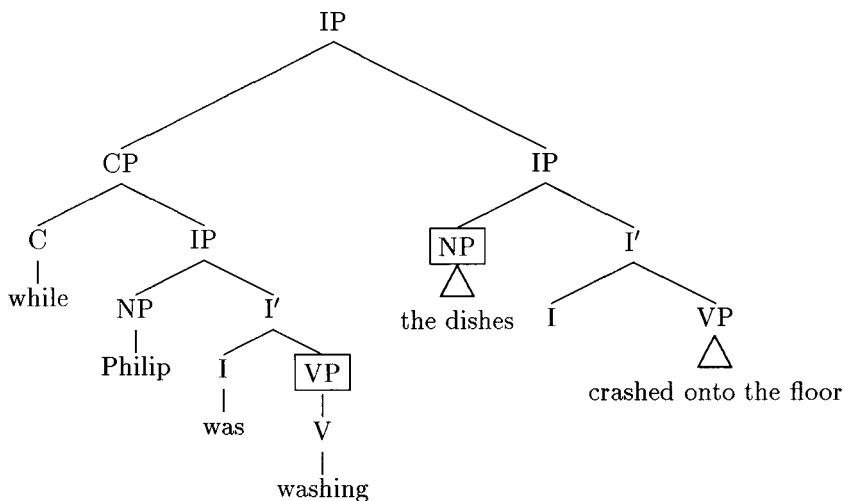
<sup>3</sup>Note the notion of “primary” versus “secondary” relations here should not be confused with “primary” versus “secondary” phrases in Construal Theory (Frazier & Clifton, 1995).

description of this, though for more detail and a range of further examples, the reader is referred to Gorrell's (1995a) original work.

Consider (2b). At the point where the parser has just received *the dishes*, this NP will have been attached as the direct object of *washing*. Thus, the set of relations will encode the fact that a VP dominates this NP (in the following diagram, we box these two nodes for clarity):



However, the following word, *crashed*, forces a reinterpretation in which the NP, *the dishes*, appears in the matrix clause:

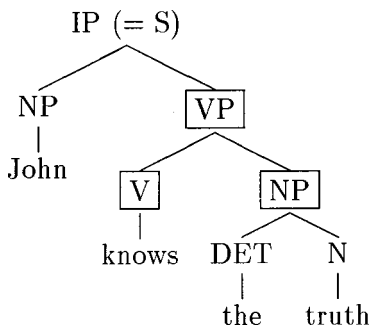


In this revised structure, it can be derived that the original VP now precedes the NP (through the inheritance condition), but this leads to a contradiction, because we now have:

$\text{dom}(\text{VP}, \text{NP}) \wedge \text{prec}(\text{VP}, \text{NP})$

against the exclusivity condition, and thus the parse is predicted to be impossible.

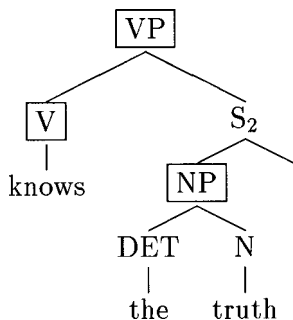
Now consider (2a). At the point where *the truth* has just been parsed, the tree-description will include an NP dominating *the truth*. This NP is dominated by VP and preceded by V:



The primary relations will include the following:

{ ...,  $\text{dom}(\text{VP}, \text{NP})$ ,  $\text{prec}(\text{V}, \text{NP})$ , ... }

Among the secondary relations, we will have, for example, the assignment of a thematic role to the direct object NP by the verb. Now, encountering the verb *hurts* will force a reinterpretation to a complement clause analysis, and a consequent revision of this assignment of thematic roles. However, the set of primary relations can be updated monotonically, with the addition of a new S node (call it  $S_2$ ).<sup>4</sup> We show the relevant section of the corresponding phrase marker below:



<sup>4</sup>In standard Government and Binding (GB) Theory, and in Gorrell's model, we would also require a new CP node immediately dominating the embedded S (or IP) node. This will mean that the secondary relations of case-assignment and government hold between V and NP before but not after reanalysis. The CP node has been omitted for clarity of exposition.

This is achieved by adding the following relations:

$$\{\text{dom}(\text{VP}, \text{S}_2), \text{prec}(\text{V}, \text{S}_2), \text{dom}(\text{S}_2, \text{NP})\}$$

The addition of the above new relations does not falsify any of the relations carried over from the previous state. For example, both  $\text{dom}(\text{VP}, \text{NP})$  and  $\text{prec}(\text{V}, \text{NP})$  are still true.

In summary, the D-theory formal machinery has allowed Gorrell to isolate a configurational level of representation (“primary relations”) which can be built up monotonically, while allowing non-monotonicity at other levels.

### Non-monotonic Semantic Interpretation

Gorrell’s original model, in common with approaches such as that by Pritchett (1992), assigns great importance to the syntactic level of representation in explaining garden path phenomena and, as such, does not concern itself with issues of semantic interpretation. Our implementation inherits this concentration on syntax. Ultimately, however, for a processing model to be plausible, we must show that the parser is capable of building semantically interpretable structures. We believe that the best approach for Gorrell’s model, as well as the model described here, is that suggested in the original D-theory paper (Marcus et al., 1983), in which the dominance relations at each point in processing are “strengthened” to immediate dominance, to form an interpretable “default” tree. To be more precise, we assume that, after each word is read and incorporated into the parse, the “closed world” assumption is applied to the description, so that, at that particular moment in processing, anything not entailed by the current set of dominance and precedence relations is assumed to be false. The result is that as each word is read, the minimal tree is available for semantic interpretation. This means that, in the case of infinite local ambiguity, even though the description will represent an infinite set of partial trees, the semantic interpreter only considers one of those trees at any one point in processing.<sup>5</sup> The non-monotonicity of secondary relations suggests that semantic interpretation must also be non-monotonic. There is no guarantee, for example, that a semantic representation available at a particular parse state will entail any representation at a subsequent state. For example, consider the following sentence:

---

<sup>5</sup>This view of semantic interpretation is similar in spirit to certain computational approaches, such as that of Shieber and Johnson (1993), which uses TAGs to pack together trees which share the same recursive structure. In this system, a semantic tree is built up simultaneously with the syntactic tree in the Synchronous TAG formalism. Default semantic values can be obtained from the semantic tree at each point in processing by assuming that no further TAG adjunctions are to be performed. Another similar approach is that of Thompson, Dixon and Lamping (1991), in which an atomic category context-free grammar is compiled into a strongly equivalent TAG.



### 3. John believed the politician was lying.

The interpretation available in a “snapshot” taken immediately after *the politician* has been read certainly does not entail the interpretation of the whole sentence.

This type of model, in which syntax is given a fairly privileged place in the processing architecture, may be contrasted with models which are concerned primarily with incrementally extracting semantic representation, and in which logical forms are built directly, without the explicit construction of a purely syntactic level of representation (e.g. Milward, 1994, 1995; Pulman, 1986). Such models derive logical forms entirely non-destructively by using logical devices such as higher-order abstraction.

### Connectedness and Incrementality

The mere existence of garden path phenomena shows that the human parser is incremental in the sense that it does not wait for disambiguating information before committing itself to an analysis of locally ambiguous material. However, given the incremental nature of syntactic processing, we are faced with a further question: How fine-grained are the units of incremental processing? In response to this question, many researchers assume a head-driven architecture, in which commitment to a syntactic analysis may only be made when a licensing head has been found in the input (Abney, 1987, 1989; Pritchett, 1992). Such a strategy implies that in a head-final language, such as Japanese for example, the processor waits until the final word of a phrase before building that phrase and thus committing itself to an analysis. However, there is experimental evidence from Dutch (Frazier, 1987) and intuitive evidence from Japanese (Inoue & Fodor, 1995) that in such languages, structuring *can* and *does* occur before the head has been encountered.

Consider the following example (Inoue, 1991, p. 102):

4. Bob<sub>ga</sub> Mary<sub>ni</sub> [<sub>*t<sub>nom/i</sub>*</sub> ringo wo tabeta]<sub>i</sub> inu<sub>i</sub> wo ageta.  
 Bob NOM Mary DAT apple ACC eat-PAST dog ACC give-PAST  
 “Bob gave Mary the dog which ate the apple”.

Comprehenders report a “surprise” effect on reaching the first verb, *tabeta* (“ate”). This is explained on the assumption that the nominative, dative and accusative arguments (“Bob”, “Mary” and “the apple”) are initially postulated as co-arguments of the same clause, in advance of reaching the verb. On reaching the transitive verb “ate”, this analysis is falsified, since the verb cannot take a dative argument. However, if, as the head-driven models would predict, the arguments are not structured in advance of the verb, but are held in some form of local memory store, then we have no simple explanation for the surprise effect. Gorrell (1995a) uses examples such as these to motivate the principle of “incremental licensing”:

*Incremental licensing*: the parser attempts incrementally to satisfy the principles of grammar.

On the other hand, no formal specification is made of what actions the parser takes if it is unable to satisfy this principle. The crucial question is that of whether the parser should be allowed to buffer constituents. For example, if the current word cannot be attached into the description under construction so as to guarantee a grammatical continuation, is it permissible to keep the word and its associated superstructure in a buffer or stack until the issue is resolved? This question is related to the issue of connectedness in that the size of the stack accessible to the parser corresponds to the amount of structure which may be left unconnected in the parser's memory (see Stabler, 1994a,b, for a discussion of this issue). Gorrell does not provide a discussion of this, though implicitly he does allow the parser to retain unconnected material in its memory, in cases where there is insufficient grammatical information to postulate a structural relation between two constituents. One example of this is the occurrence of a sequence of two non-case-marked NPs, as in the following centre-embedded example:

5. The man the report criticised was demoted.

On this question, Gorrell (1995a) claims that "there is no justification for asserting any relation between the two NPs". This implies that the parser is able to store unconnected material in its memory. If this is so, then it is necessary to constrain the conditions under which material may be added to this store. In particular, if the parser is permitted to shift material onto a stack whenever it fails to make an attachment, then, in a garden path utterance, the error will not be recognised until the end of the input has been reached, when the parser will be faced with a stack full of irreducible structures. This was essentially the problem faced by Abney (1987), whose model required the use of an unbounded stack. This was because attachment could only be made under a head-driven form of *licensing*, so that in left-branching structures, where the licensor was still unread in the input, it was essential to shift structure onto the stack until the licensor was found, in order for the attachment to be made. The problem was that the parser could not tell whether to continue adding material to the stack, in the expectation of a licensor later in the input, or whether to abandon the parse. In a later version (Abney, 1989), this problem was solved with the addition of LR-states, to indicate whether or not a grammatical continuation could be expected at the current input.

For the purposes of the implementation reported in this paper, we have taken the most constrained position possible, and insisted on full connectedness; that is, at any stage, the parser has access only to a single set of relations describing a fully connected tree, and each word has to be incorporated within this structure as it is encountered. Behind the adoption

of this strict regime lies the acceptance of the following hypothesis, various forms of which have appeared in the work of several researchers in the field (Stabler, 1994a,b; Steedman, 1989):

Every structure associated with every prefix of a readily intelligible, grammatical utterance is connected (Stabler, 1994a).

## CONSTRAINTS ON THE MODEL

In this and subsequent sections, we describe our own implemented model. We begin by making explicit the constraints on which the model is based, and go on to formulate the attachment operations used by the parser, and discuss the empirical consequences of the model.

The implementation described here obeys the following constraints, which are intended to capture, and build on, the conditions described in Gorrell's work. In particular, informational monotonicity is defined, as well as full specification. The condition of incrementality is stronger than that implied by Gorrell, since it insists on full connectedness.

1. *Strict incrementality*: each word must be connected to the current tree description *at the point at which it is encountered* through the addition of a non-empty set of relations to the description.
2. *Structural coherence*: at each state, the tree description should obey the conditions on trees: (a) single root condition; (b) exclusivity condition; (c) inheritance.
3. *Full specification of nodes*: tree-descriptions are built through the assertion of dominance and precedence relations between fully specified nodes. In the current implementation, each node is a triple  $\langle \text{Cat}, \text{Bar}, \text{Id} \rangle$ , consisting of category *Cat*, bar-level *Bar* and an identification number *Id*. Each of these three arguments must be fully specified once the structure has been asserted.<sup>6</sup>
4. *Informational monotonicity*: the tree-description at any state *n* must be a subset of the tree-description at state  $n + 1$ .<sup>7</sup> Thus the parser may not delete relations from the tree description.
5. *Obligatory assertion of precedence*: if two or more nodes are introduced as sisters, then precedence relations between them must be specified.

---

<sup>6</sup>As an anonymous reviewer has pointed out, the insistence on full specification of syntactic category is almost certainly too strong; it does not allow us to capture the cases in which the part of speech of a word is ambiguous, and resolved only after subsequent input has been read, with little appreciable processing cost (Frazier & Rayner, 1987). Though we have not worked out the details, a fuller account might allow category names to be changed, while retaining certain aspects of the basic structural skeleton.

<sup>7</sup>In fact, since (1) requires the set of relations added to the description at each word to be non-empty, the description at *n* is a *proper* subset of the description at  $n + 1$ .

6. *Grammatical coherence*: at each state, each local branch of the phrase marker described must be well-formed with respect to the grammar.<sup>8</sup>

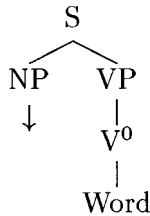
## COHERENCE-PRESERVING PARSING OPERATIONS

### Syntactic Representation

The syntactic representation assumed is similar to that used in tree-adjoining Grammars (henceforth TAGs) (Joshi et al., 1975) and, in particular, *lexicalised* TAGs (Schabes et al., 1988). Each lexical category is associated with a set of structural relations, which determine its *lexical subtree*. For example, the verb category for English contains the following relations:

$$\{ \text{dom}(\text{VP}, \text{V}^0), \text{dom}(\text{V}^0, \text{Word}), \text{dom}(\text{S}, \text{VP}), \\ \text{dom}(\text{S}, \text{NP}), \\ \text{prec}(\text{NP}, \text{VP}) \}$$

These define the following subtree, where we borrow a tradition from the TAG literature and represent an attachment site with a downward-pointing arrow:



As each word is encountered in the input, the parser attempts to add the set of structural relations defining that word's lexical subtree to the set of structural relations defining the global representation under construction, in a way which is similar to the "Assertion Set" approach to parsing (Barton & Berwick, 1985). There are two possibilities for this, simple attachment and tree-lowering, corresponding roughly (though not exactly, as we shall see later) to substitution and adjunction in TAG.<sup>9</sup>

---

<sup>8</sup>Note that adopting the grammatical coherence constraint does not render strict incrementality vacuous. Strict incrementality requires full connectedness, and no buffering of input, while grammatical coherence is intended to guarantee that the (connected) "default tree" produced at each incremental point in processing is specified enough to support semantic interpretation, at least in principle.

<sup>9</sup>Note that the parser is not *head-driven*, since, for example, we allow lexical subtrees to include material not dominated by the word's maximal projection. In a head-final construction, this allows arguments and adjuncts to incorporate themselves into the projection of the licensing head before this head is reached in the input. Empty heads do not play any structure-building role in the parser.

## Simple Attachment

The parser is capable of performing simple right and left attachment, illustrated schematically in Fig. 1. A lexical subtree may contain attachment sites to the left or the right of the word from which it is projected. These are distinguished empty nodes, which must be filled in accordance with the conditions on trees (the parser effectively keeps these nodes on a stack; see Appendix for details). Intuitively, left attachment consists in attaching the current global tree onto the left corner of the subtree projection of the new word, while right attachment consists in attaching the subtree projection of the new word onto the right corner of the current global tree. This is done by simply identifying the root node of one subtree with an attachment site from the other subtree, and adding the required structural relations to the global representation. The attachment operations are similar to Abney's (1987, 1989) *Attach-L* and *Attach*, respectively. In the following definitions, we use the term "current tree description" to refer to the set of relations describing the global phrase-marker currently in the parser's memory; in other words, the parser's *left-context*. The term "subtree projection" is used to refer to the set of relations corresponding to the lexical category of the new word encountered in the input.

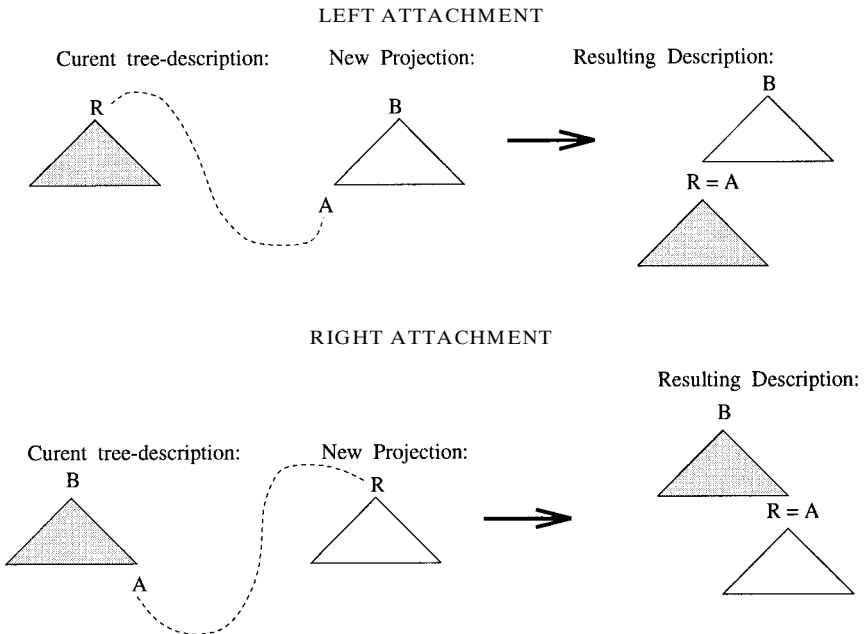


FIG. 1. Left and right attachment.

- *Left attachment*: let  $D$  be the current tree description, with root node  $R$ . Let  $S$  be the subtree projection of the new word, whose left-most attachment site,  $A$ , is of identical syntactic category with  $R$ . The updated tree description is  $S \cup D$ , where  $A$  is identified with  $R$ .
- *Right attachment*: let  $D$  be the current tree description, with the first right attachment site  $A$ . Let  $S$  be the subtree projection of the new word, whose root,  $R$ , is of identical syntactic category with  $A$ . The updated tree description is  $S \cup D$ , where  $A$  is identified with  $R$ .

The parser is also capable of creating a new attachment site with reference to a verb's argument structure. For example, if a transitive verb is found in the input, then a new right attachment site is created for a NP, and a new NP node is "downwardly projected" as a sister to the verb.<sup>10</sup>

So, for a simple transitive sentence such as *Polly eats grapes*, first *Polly* is projected to a NP and instantiated as the current global tree. This NP will match the left attachment site of *eats*, and so left attachment will be performed. Since *eats* is a transitive verb, a new NP attachment site will be created, and this will be suitable for right attachment of the projection of *grapes*.

If simple attachment is not possible, then the parser attempts to perform a second mode of attachment, *tree-lowering*. Before describing tree-lowering, we will briefly review the adjunction operation of TAGs, since the two operations share some common features.

## TAG Adjunction

A TAG contains a set of elementary trees, which are divided into *initial* and *auxiliary* trees. An initial tree is rooted in a distinguished *start symbol* non-terminal node, and has terminal symbols along its frontier. An auxiliary

---

<sup>10</sup>Note that this requires a systematic method for choosing among alternative subcategorisation frames in cases where the verb is ambiguous. Gorrell (1995a) defines the notion of simplicity: "No vacuous structure building"—the principle which is used to explain the initial preference for NP attachment in the case where a verb may subcategorise either for a NP or for a clause. In the case of an optional argument, however, the processor cannot employ downward projection, since there is no guarantee whether or not the argument will appear in the input. This implies that, if the parser is able to cope with the attachment of optional arguments, it must be capable of attaching an argument without first downwardly projecting that argument. Thus, it may be possible to avoid the use of downward projection altogether. One option for dealing with this would be to project complements "on demand", when their presence becomes indicated by the input. Note that taking such an action may require an argument or theta-attachment preference to be stipulated in the parser, since this would no longer be guaranteed by the "simplicity" of attaching material into a predicted attachment site (Gorrell, 1995a). This would also allow us to replicate the *delay* in the use of subcategorisation information proposed by Mitchell (1989), as opposed to the *immediate* use which downward projection implies.

tree is rooted in a non-terminal node, and its frontier includes a further non-terminal node, which must be of the same type as the root. This non-terminal node is known as the foot node.

Most versions of TAG allow two operations for combining one tree with another: substitution and adjunction. Substitution is essentially the same operation as simple attachment, as defined above. To explain adjunction, we refer to an example instantiation given in Fig. 2.

The following definition is adapted from Joshi, Vijay-Shanker and Weir (1991). Let  $\alpha$  be an initial tree containing a node  $n$  with category  $c$  (this is the node marked “ $c:n$ ” in Fig. 2). Let  $\beta$  be an auxiliary tree, whose root and foot nodes are also of category  $c$ . The adjunction of  $\beta$  to  $\alpha$  at node  $n$  will yield the tree  $\gamma$  that is the result of the following operations:

1. The subtree of  $\alpha$  dominated by  $n$  is excised. We call this excised subtree  $t$ .
2. The auxiliary tree  $\beta$  is attached at  $n$ , and its root node is identified with  $n$ .
3. The excised subtree  $t$  is attached to the foot node of  $\beta$ , and the root node,  $n$  of  $t$ , is identified with the foot node of  $\beta$ .

It is simple to define a version of TAG adjunction in terms of sets of structural relations which describe trees. The trees  $\alpha$ ,  $\beta$  and  $\gamma$  can be described by sets of relations, which we call  $A$ ,  $B$  and  $\Gamma$ , respectively. For example,  $A$  (which describes  $\alpha$ ), is as follows (again, we use  $c:n$  to represent the node  $n$  with category  $c$ ):

$$A = \{\text{dom}(a,b), \text{dom}(a,c:n), \text{dom}(a,d), \text{dom}(c:n,g), \\ \text{dom}(c:n,h), \text{prec}(b,c:n), \text{prec}(c:n,d), \text{prec}(g,h)\}$$

The set  $B$  is similarly defined. Now, to adjoin  $\beta$  to  $\alpha$ , we find the set of local relations  $L \subseteq A$  in which  $c:n$  participates:<sup>11</sup>

$$L = \{\text{dom}(a,c:n), \text{prec}(b,c:n), \text{prec}(c:n,d)\}$$

We then build a new set of relations  $N$ , which is  $L$  with all occurrences of  $c:n$  replaced by the root node of  $\beta$  (call it  $c:r$ ):

$$N = \{\text{dom}(a,c:r), \text{prec}(b,c:r), \text{prec}(c:r,d)\}$$

We then identify the foot node of  $\beta$  with  $c:n$ , so that  $B$  consists of the following set of relations:

$$Y = \{\text{dom}(c:r,e), \text{dom}(c:r,c:n), \text{dom}(c:r,f), \text{prec}(c:n,f)\}$$

Now  $\Gamma$  is simply defined as follows:

---

<sup>11</sup>The “local relations” in which a node  $N$  participates at state  $S$  are those dominance and precedence relations which define the mother and sisters of  $N$  at  $S$ .

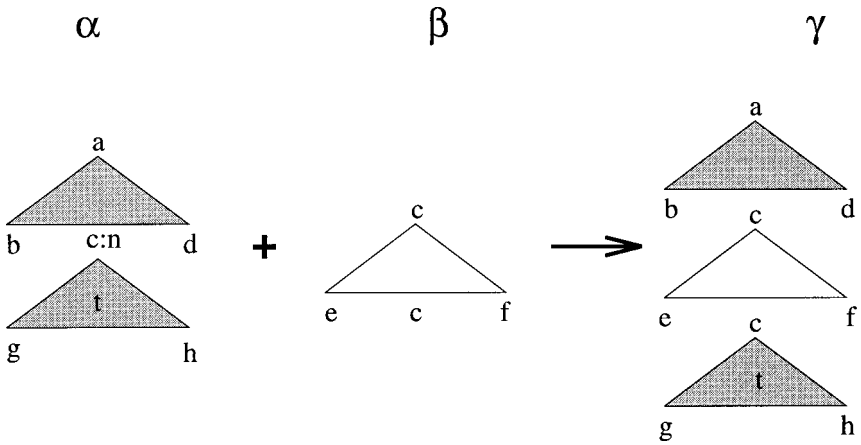


FIG. 2. TAG adjunction.

$$\Gamma = A \cup N \cup B$$

As the reader can verify, the derived set  $\Gamma$  describes the tree  $\gamma$  as required.

### Tree-lowering

In standard TAG adjunction, the auxiliary node (that is, the node at which adjunction is performed;  $c:n$  in the above example) is, informally speaking, “split” into a bottom half, which is identified with the foot node of the auxiliary tree, and a top half, which is identified with the root node of the auxiliary tree. However, in the version of adjunction which we have defined above, in terms of sets of relations, we identify  $c:n$  only with the foot node of the auxiliary tree, and treat the root node of the auxiliary tree ( $c:r$  in the above example) as a separate node that takes over the place previously occupied by  $c:n$ . One consequence of this is that we are able to generalise the adjunction operation to include cases where the root and foot nodes are of distinct syntactic categories, which is impossible in a standard TAG.<sup>12</sup> It is this property which allows us to capture reanalysis phenomena in a monotonic fashion, since it provides a way of replacing node  $X$  with node  $Y$ , while also guaranteeing that the original position of  $X$  dominates the revised position of  $X$ .

Consider example (1a), repeated below as (6):

<sup>12</sup>Vijay-Shanker (1992) discusses a version of TAG in which root and foot nodes need not be of the same category. However, Vijay-Shanker’s system differs from that proposed here in that it explicitly divides auxiliary nodes into a top and bottom half, into which structure can subsequently be inserted, and is intended to capture, grammatical rather than processing phenomena (see Sturt and Crocker, 1995, for a fuller discussion).



## 6. John knows the truth hurts.

Schematically, what we need in the above case is illustrated in Fig. 3. Here, the nodes corresponding to root and foot nodes of the auxiliary tree are the S node and NP (subject) node of the embedded clause, respectively. In order to accommodate  $\beta'$  into  $\alpha'$  in the desired monotonic fashion, therefore, we will have to drop the requirement for root and foot nodes to be of identical syntactic category. However, we must constrain this operation so that it may only be employed in cases where the root node of the auxiliary tree is licensed in its new adjoined position. In the above case, it *is* licensed, since *know* may subcategorise for a clause.<sup>13</sup>

In order to maintain structural coherence, the new word attached via tree-lowering must be preceded by all other words previously attached into the description. We can guarantee this by requiring the lowered node to dominate the last word to be attached. We also need to ensure that, to avoid crossing branches, the lowered node does not dominate any unsaturated attachment sites (or “dangling nodes”). In other words, in order to obey the theory of trees, the node selected for tree-lowering must be accessible in the following sense:<sup>14</sup>

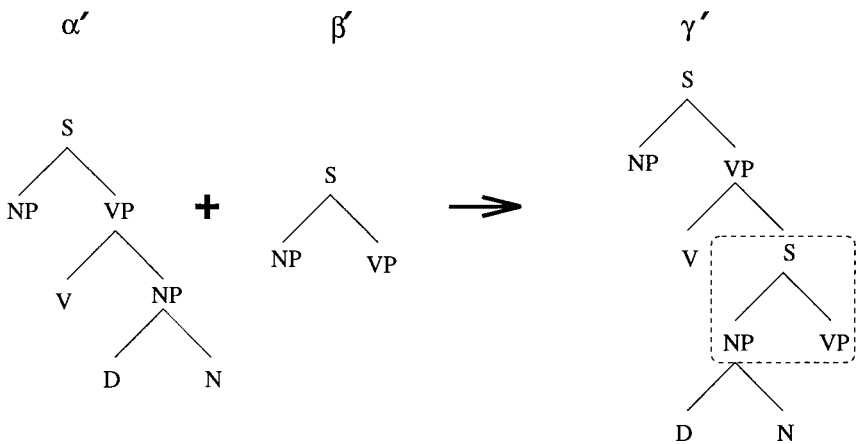


FIG. 3. Reanalysis as the insertion of one tree inside another. The inserted material is enclosed inside the dotted lines.

<sup>13</sup>In the current implementation, lexical information for the verb *know* is re-accessed in order to find the alternative subcategorisation frame.

<sup>14</sup>Though accessibility can be derived from the theory of trees, the actual parser does not reason from first principles in order to find accessible nodes, but explicitly uses the above definition. This is done for efficiency reasons.

DEFINITION: *Accessibility*

Let  $N$  be a node in the current tree description. Let  $W$  be the last word to be attached into the tree.

$N$  is accessible iff  $N$  dominates  $W$ , and  $N$  does not dominate any unsaturated attachment sites.<sup>15</sup>

Note that it is not necessarily the case that all nodes on the right edge of the tree are accessible, nor that all accessible nodes are on the right edge of the tree. For example, a “dangling node” will not dominate any lexical material, even though it might be on the right edge of the tree, and therefore it will not be accessible. Also, there may be a node which dominates the last word to be attached, and which is therefore accessible, but which precedes a dangling node, and is therefore not on the right edge of the tree.

The tree-lowering operation is defined as follows, and illustrated diagrammatically in Fig. 4:

DEFINITION: *Tree-lowering*

Let  $D$  be the current tree description. Let  $S$  be the subtree projection of the new word. The left attachment site  $A$  of  $S$  must match a node  $N$  accessible in  $D$ . The root node  $R$  of  $S$  must be licensed by the grammar in the position occupied by  $N$ . Let  $L$  be the set of local relations in which  $N$  participates. Let  $M$  be the result of substituting all instances of  $N$  in  $L$  with  $R$ . The attachment node  $A$  is identified with  $N$ .

The updated tree-description is  $D \cup S \cup M$ .

## TREE LOWERING

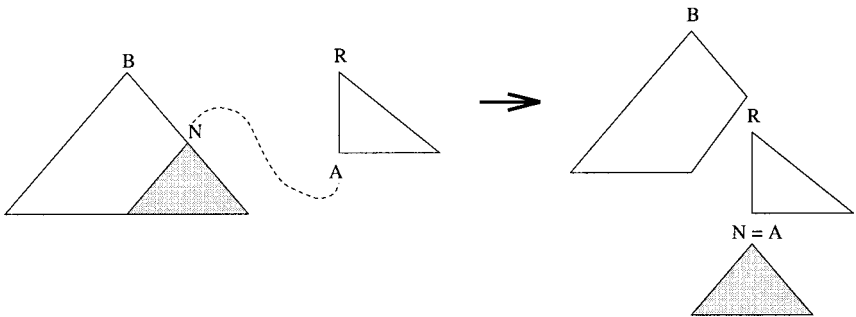


FIG. 4. Schematic illustration of tree-lowering. The node  $R$  must be licensed in the position previously occupied by  $N$ .

<sup>15</sup>Note that tree-lowering constrained by accessibility results in a system which is very similar to that of Stevenson's (1993, 1994) competitive attachment model. See pp. 487–488 for a discussion.

Note that, in order to check whether the root node *R* of the new subtree projection is licensed in its new position, it may be necessary to access subcategorisation information associated with a word long past in the input, and which is no longer *accessible* in the sense defined above. This is the case in the above example, where the subcategorisation frame of *knows* has to be checked to allow the attachment of the clausal node as its sister, although the  $V^0$  node of *knows* itself is no longer accessible. It would be interesting to investigate how far inside the tree the parser is capable of looking in order to extract this type of information.<sup>16</sup>

The parser is constructed in such a way that, if at any point in the parse simple attachment fails, the accessible nodes of the current tree-description are considered until a node is found at which tree-lowering may be applied. Note that tree-lowering can capture many effects of standard TAG adjunction, and it is therefore possible to use this operation for the attachment of post-modifiers, which is the course of action taken in this implementation. The preference for argument over adjunct attachment is captured by the fact that tree-lowering is only attempted in cases where standard attachment fails.<sup>17</sup> Note, however, that we do not claim that adjunct attachment exhibits the same cost as reanalysis. The distinction is that adjunct attachment involves only *adding* a new dependency, while reanalysis crucially involves *breaking* a dependency<sup>18</sup> (see pp. 484–487 for a discussion).

## TOP-DOWN PREDICTION

It was mentioned earlier that, for head-initial languages like English, when the parser encounters a head requiring a following internal phrasal argument, this argument is projected top-down and asserted as a so-called “dangling node”. However, there will be cases where the word immediately following the head cannot be connected directly to such a dangling node. Consider the following example:

### 7. Mary thinks John ...

On encountering the verb *thinks*, the parser must project a clausal node, since this verb can only subcategorise for a clause. However, the NP *John* cannot be connected directly to this node, since it is of the wrong syntactic type. The current implementation addresses this problem by adopting Crocker’s (1992, 1994) approach, in which the “functional structure” of the clause (CP, IP) is projected top-down along with the NP subject node in the

<sup>16</sup>We are grateful to Martin Pickering for bringing this point to our attention.

<sup>17</sup>In Gorrell’s original model, this is accounted for by the Principle of Simplicity (Gorrell, 1995a).

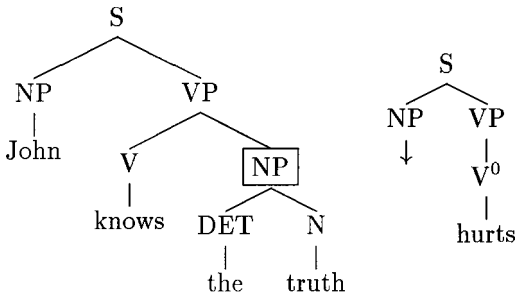
<sup>18</sup>In Gorrell’s (1995a) terms, reanalysis involves deleting secondary relations.

specifier position of IP. This provides an immediate attachment site for the embedded subject. This is an example of the parser's "non-lexical structure building", as discussed in the Appendix.<sup>19</sup>

## SEARCH STRATEGIES FOR COHERENCE-PRESERVING REANALYSIS

### English

In example (6), at the point where standard attachment fails, the parser is faced with the task of incorporating the projection of *hurts* into its representation. In this case, there is only one accessible node at which tree-lowering may be applied in such a way that grammatical licensing conditions are met, and that is the NP boxed in the diagram below:



In the above, then, the word *hurts* uniquely disambiguates the local ambiguity. However, there may be occasions where reanalysis does not coincide with unique disambiguation. In particular, the parser may be faced with a choice of alternative lowering sites in much the same way as it is faced with a choice of alternative attachment sites at the onset of a standard local ambiguity, and therefore we must consider the search strategy, or heuristics, used to choose between such lowering sites. For example, imagine that the following utterance has just been processed:

8. I know [<sub>NP<sub>1</sub></sub> the man who believes [<sub>NP<sub>2</sub></sub> the countess]]

Now, imagine that the utterance continues with the verb *killed*. The verb must be attached via tree-lowering, but now there are two accessible nodes where the operation can be applied: NP<sub>1</sub> and NP<sub>2</sub>. Though, as far as we are aware, no experimental studies of this type of structure have been conducted, it intuitively seems that the lower site, NP<sub>2</sub>, is preferred. This can be seen more clearly in the following sentences, where binding constraints

<sup>19</sup>Milward's (1994, 1995) rule of *state prediction* gives a more general solution to this problem using a dynamic grammar formalism (see Sturt and Crocker, 1995, for further discussion on this point).

force a particular reading: (9a), where lowering is obligatorily applied at the lower NP, is easier than (9b), where lowering can only be applied at the higher NP.

- 9a. I know the man who believes the countess killed herself.  
 b. I know the man who believes the countess killed himself.

In the above examples, though the verb *killed* triggers reanalysis, it is the following reflexive pronoun, *himself/herself*, which uniquely disambiguates the structure. If it is indeed the “low” reanalysis [corresponding to (9a)] that is favoured, then in the dispreferred case, (9b), the parser will *mis*-reanalyse on encountering *killed*, only to experience what we might think of as a “second-order” garden path effect at the disambiguating signal, *himself*, where the preferred reanalysis is seen to have been mistaken.

If the preference to reanalyse at a low, more recent site does indeed exist, it may be compared with a similar preference to attach post-modifiers to low sites, which follows from principles such as Right Association (Kimball, 1973; Phillips, 1995) or Late Closure (Frazier, 1978). Thus, for example, the preferred reading for *John said Bill left today* is (10a), where the adverbial appears in the lower clause, as opposed to (10b), where it appears in the higher clause:

- 10a. John said [Bill left today]  
 b. John said [Bill left] today

Since tree-lowering is also used for post-modifier attachment, we would expect the search strategy used in examples such as (9a) to share some features in common with that in (10a). A possible strategy which can be used, then, is to search the set of accessible nodes in a bottom-up direction. We define the *current node path* as the ordered set of accessible nodes. To implement the bottom-up search, the current node path is ordered from bottom to top, so that, in the path  $\langle N_1, N_2, \dots, N_n \rangle$ , the node  $N_1$  is the node immediately dominating the last word to be processed (i.e. the lowest accessible node),  $N_n$  is the highest accessible node, and  $N_i$  immediately dominates  $N_{i-1}$  for each pair of adjacent nodes in the path. The parser will then consider, in the order given, each node in the path as a possible lowering site.<sup>20</sup>

In the bottom-up search, then, the parser considers the first node in the path,  $N_1$ , and attempts to lower. If this is unsuccessful, it moves to the next node,  $N_2$  (i.e. the node immediately dominating  $N_1$ ), and again attempts to lower. The process continues, with the parser considering successively higher nodes until either lowering is successful, and the parser can move on to consider the next word, or the current node path is exhausted, in which

<sup>20</sup>Note that different search strategies may be implemented by changing the ordering function, and this will be exploited in the next section to deal with Japanese.

case the parser fails, and the string is rejected as either a conscious garden path or ungrammatical.

It will be noted that the simple bottom-up search reflects a preference to lower the most recent node possible. However, we recognise that recency is not the only factor in the attachment of post-modifiers (Cuetos & Mitchell, 1988; Gibson, Pearlmutter, Canesco-Gonzalez, & Hickok, 1996), so that the pure bottom-up search we propose here can only be seen as an approximation. Later, we discuss possible factors which might influence the selection of a site for tree-lowering (see pp. 484–487).

### The Restrictiveness of the Model

The reader can verify that the attachment and lowering operations as defined above are guaranteed to preserve the conditions on trees, and thus to satisfy the monotonicity criterion (Gorrell's "structural determinism"). For the attachment case, this is trivially true. For the lowering case, it can easily be seen that the source position of the lowered node (i.e. its configurational position before the lowering operation is applied) is guaranteed to dominate its target position (*after* lowering has been applied). The result of this is that the present model is at least as restrictive as Gorrell's model; that is, all constructions predicted as conscious garden paths in Gorrell's model are also ruled out in the present model (i.e. they do not receive a parse).

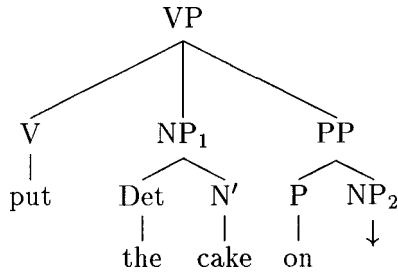
Consider the following well-known conscious garden path example:

11. John put the cake on the table in his mouth.

Assume that the first PP *on the table* is initially attached as the location argument of the verb *put*. In the present model, as soon as the preposition *on* is attached under the projected PP node, all nodes preceding the PP are "closed off" from accessibility. In particular, no adjunction within NP<sub>1</sub> will be possible, since NP<sub>1</sub> will never dominate "the last word to be processed" at this or any other subsequent point in the parse. The result of this is that a second PP [*in his mouth* in (11)] can subsequently be attached as a modifier of *the table*, but that the revisions required to recover the plausible reading (i.e. re-attaching the first PP, *on the table*, as a modifier of the direct object, *the cake*, and attaching the second PP, *in his mouth*, as the new locative argument of *put*) will not be possible.<sup>21</sup>

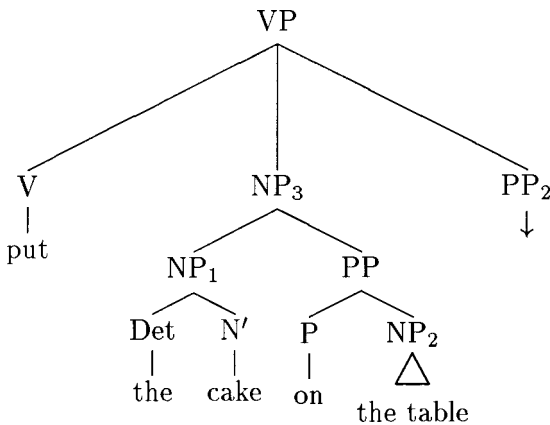
---

<sup>21</sup>This prediction is supported by on-line experimental evidence. Adams (1995) reports that reanalysis of the type required for (11), which she calls "theme repair", is more difficult than other options for attaching the second PP, including attachment of the second PP as a modifier of *the table*.



In Gorrell's model, this restriction is captured by the fact that the precedence relation between  $NP_1$  and PP would have to be altered if PP were to be subsequently reanalysed as an adjunct within  $NP_1$  (otherwise,  $NP_1$  would simultaneously dominate and precede PP).

As an aside, it is worth noting that in the present model, once PP is attached in its argument position, no adjunction is possible, either to  $NP_1$  itself, or to any node dominated by  $NP_1$ , whereas in Gorrell's model, adjunction is ruled out to non-maximal projections *properly dominated*<sup>22</sup> by  $NP_1$ , but not to  $NP_1$  itself (i.e. the maximal projection). The following diagram illustrates the adjunction of PP to the maximal projection  $NP_1$ . It can be seen that in Gorrell's model, the adjunction can be performed simply by adding a new NP node (which we index  $NP_3$ ), which dominates  $NP_1$  and PP. This is because the precedence relation between  $NP_1$  and the PP node in its original argument attachment site still holds after the adjunction of PP to  $NP_1$ . In contrast, the parsing model outlined in this paper cannot perform this adjunction once the PP has been attached in its argument position, because, as explained above,  $NP_1$  becomes inaccessible to tree-lowering as soon as the argument attachment of PP has been made.



<sup>22</sup> $X$  properly dominates  $Y$  iff  $X$  dominates  $Y$  and  $X \neq Y$ .

Thus, in order to account for the difficulty of such examples, Gorrell would need to rule out the use of this type of adjunction explicitly, either by appealing to the competence theory, or by placing a constraint on the parser forbidding the “extension” of a maximal projection in this way. This is important, because other well-known garden path examples are predicted to be difficult in Gorrell’s model for essentially the same reasons (i.e. because the precedence relation between an NP and a following phrase has to be “converted” to a dominance relation), and would therefore also rely on ruling out this type of adjunction. Examples include *The horse raced past the barn fell* and *The psychologist told the woman he was having trouble with to leave*. We do not necessarily wish to argue for or against the NP adjunction hypothesis on the competence level,<sup>23</sup> but rather we use this as an example to show how the model proposed here is essentially more restrictive than Gorrell’s in its range of predictions.

Finally, we would like to point out a class of examples for which both Gorrell’s model and the model proposed here under-predict. Gorrell (1995a, ch. 5) points out that a model constrained by structural determinism must allow *weak interaction* of non-syntactic knowledge, in the sense of Crain and Steedman (1985). That is, at each point in processing, it must be possible for discourse information and real-world knowledge to be used in order to choose among different alternative analyses proposed by the syntactic processor. This means the model faces problems in cases where the implausibility of an initially preferred analysis forces *retrospective* reanalysis.

Consider the examples in (12):

- 12a. John saw the man with a telescope.  
 b. John saw the man with a moustache.

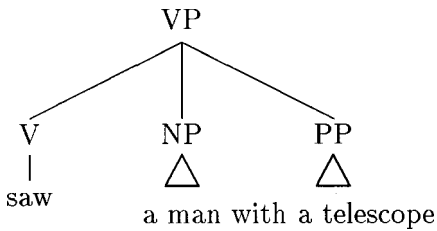
Assume that there is a preference for the instrumental reading, in which the PP is attached as an argument to the verb (Rayner, Carlson, & Frazier, 1983). The constraint of incrementality means that the argument attachment of the PP must be made as soon as *with* is encountered. In the case of (12a), this analysis remains plausible, since a telescope is a reasonable instrument with which to see a man. In the case of (12b), however, the instrumental analysis is eventually found to be implausible, since moustaches are, in general, rather poor optical instruments, and *with a moustache* will have to be reattached as an adjunct of the NP *a man*. We show the two analyses in (13) and (14):

---

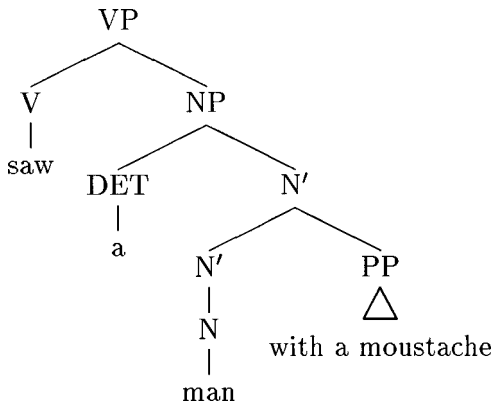
<sup>23</sup>Though the existence of sentences such as *The sweet and the cake on the table looked delicious*, where both NPs *the sweet* and *the cake* seem to be modified by the PP *on the table*, make this a reasonable assumption.



13.



14.



On encountering the word *moustache* in (12b), the parser will have to reanalyse from the structure in (13) to that in (14). But this will result in an incoherent description, since in (13) we have  $\text{prec}(\text{NP}, \text{PP})$ , and in (14) we have  $\text{dom}(\text{NP}, \text{PP})$ , against the exclusivity condition, and yet, though on-line preferences are known, neither of the two sentences given in (13) or (14) seems to cause conscious processing difficulty.<sup>24</sup> The difficulty is mirrored in the present model by the fact that the NP, along with all the nodes it dominates, becomes inaccessible for tree-lowering as soon as the preposition *with* is attached.

There are various possibilities for allowing both (12a) and (12b) to be processed [corresponding to the structures in (13) and (14)], and we are currently working on a revised underspecification formalism which does not rely so heavily on purely configurational information. This is a non-trivial problem, because we want the model to predict not only the ease of the sentences in (12), but also the difficulty of the structurally similar (11).

<sup>24</sup>The reader can verify that a binary branching variant of (13) will suffer the same problem.

## PROCESSING JAPANESE

In the previous section, we saw that, at the point of reanalysis, the parser may be faced with a choice of possibilities at which the lowering operation may be applied, necessitating the definition of a search strategy. In this section, we look at a class of examples involving the reanalysis of relative clauses in Japanese, where just such a choice is found.

We will first look at Gorrell's explanation for the distinction between conscious and unconscious garden paths for this type of example in terms of a comparison between the set of relations describing the phrase marker at two snapshots of processing. We then show how this "static" explanation fails to make an adequate distinction between the easy and hard cases of reanalysis. We subsequently demonstrate the important role of the search strategy in explaining this distinction, showing, incidentally, how the bottom-up strategy we proposed to account for the English data predicts the opposite of the observed results. We then give a (speculative) proposal of a top-down, weakly interactive search, which allows a more satisfactory explanation.

The issue concerns data such as the following:

- 15a. [Mary ga sinseihin wo  $t_{loc}$  kaihatu<sub>sita</sub>] kaisy<sub>a<sub>i</sub></sub> ga tubereta.  
 Mary NOM new product ACC developed company NOM went bankrupt  
 "The company where Mary developed the new product went bankrupt".  
 (Inoue, 1991)
- b. Yamasita ga [ $t_{nom/i}$  yuuzin wo houmonsita] siriai<sub>i</sub> ni tegami  
 yamasita NOM friend ACC visited acquaintance DAT letter  
 wo kaita.  
 ACC wrote  
 "Yamasita wrote a letter to an acquaintance who visited his friend".  
 (adapted from Mazuka & Itoh, 1995)
- c. Yamasita ga yuuzin wo [ $\emptyset_{nom} t_{acc/i}$  houmonsita] kaisy<sub>a<sub>i</sub></sub> de mikaketa.  
 Yamasita NOM friend ACC visited company LOC saw  
 "Yamasita saw his friend at the company he visited".  
 (adapted from Mazuka & Itoh, 1995)

In all of the examples in (15), a clause is initially built containing an (overt) subject, object and transitive verb. However, subsequent appearance of the noun shows that this clause, or some part of it, must be reinterpreted as a relative clause modifying that noun. The local ambiguity consists in the fact that the boundary between the main and relative clause may fall at any point between the left edge of the sentence and the immediately pre-verbal position (Japanese is a "super pro-drop" language). We will see how this local ambiguity can be modelled as a choice of lowering sites at the point

where the parser receives the immediately post-clausal noun.<sup>25</sup> In each of the three sentences in (15a–c), the initial string may be represented schematically as follows:<sup>26</sup>

NP<sub>nom</sub> NP<sub>acc</sub> V<sub>trans</sub>

On the assumption that the two NPs are initially structured as co-arguments of the verb, the parser may have to “displace” one or more arguments from the clause on reaching the relativising noun. This is because a relative clause must contain a gap, which is co-indexed with the head noun, and if the constituent to be relativised is already overt in the initially built clause, then that overt constituent will have to be displaced from the clause, and replaced with the gap. However, if the relativised constituent is not overt in the initially built clause, then no material will have to be displaced. For example, if the relativised constituent is represented by an empty *pro* category in the initially built clause, then the relativisation relation can be established by postulating the *pro* as the relativised trace, and co-indexing it with the head noun. Otherwise, if the relativised constituent is a non-overt adjunct in the initially built clause, then the relativisation relation can be established by adding the empty category representing the adjunct to the clause. Inoue (1991) notes a general preference towards displacing the minimal amount of material from a completed clause to a higher clause. He calls this the “Minimal Expulsion Strategy”.<sup>27</sup>

The sentence in (15a) is an example of a case where no material is displaced on reaching the head noun. This is because, although both arguments of the verb *kaihatu* are present in the structure, the relativised constituent corresponds to a locative *adjunct* rather than an argument, so no overt constituents need to be displaced by postulating the relativised gap. Note that, in Japanese, relativisation of an adverbial is only possible if the

---

<sup>25</sup>At the point where a post-clausal noun has been found, the parser is actually faced with a further local ambiguity; either the clause directly modifies the post-clausal noun through relativisation, or the post-clausal noun is the first word of a new relative clause which modifies a subsequent head noun in coordination with the first relative clause. However, Mazuka et al. (1989) show that, for strings which are globally ambiguous for these two readings, the reading consistent with a direct association between a relative clause and a following noun is strongly preferred to the indirect “coordinate” reading, even in cases where plausibility favours the coordinate reading. Inoue (1991), Inoue and Fodor (1995) and Fodor and Inoue (1994) report cases where an initial preference for “direct” reading results in an unrecoverable garden path effect when this is later found to be untenable.

<sup>26</sup>We indicate an NP bearing case *C* as NP<sub>*C*</sub>, and V<sub>trans</sub> denotes a transitive verb (i.e. a verb which takes one nominative and one accusative argument).

<sup>27</sup>We prefer to use the term “displacement” over “expulsion”, since as we shall see in the present model, a displaced element is not expelled from the clause to which it is originally attached, but rather everything except the displaced element is lowered. However, we continue to use the term “minimal expulsion” when we refer to the processing strategy as proposed by Inoue.

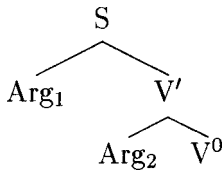
role of that adverbial is temporal or locative. We will refer to such examples as “null displacement”. Null displacement examples of this kind reportedly do not cause conscious processing difficulty (Inoue, 1991).

The sentence in (15b) involves displacing *one* argument (i.e. the nominative NP *Yamasita ga*). At the point where the first verb is processed, a clausal structure will have been built, corresponding in meaning to “Yamasita visited his friend”. However, on the subsequent input of the noun *siriai*, a gap has to be found in that clause. This time, since *siriai* (“acquaintance”) is not a plausible location or time for an action to take place, adverbial relativisation is not possible (consider the bizarre interpretation of the English NPs “the acquaintance *where* Yamasita visited his friend”, and “the acquaintance *when* Yamasita visited his friend”). This means that the processor will be forced to postulate a gap in one of the two argument positions. Postulating the gap in the subject position causes the displacement of *Yamasita ga*, which results in the globally correct structure. We will call such examples “single displacement” sentences. They do not cause conscious processing difficulty (Mazuka & Itoh, 1995).

The sentence in (15c) involves displacing *two* arguments, the subject argument and the object argument. We will postpone our discussion of how processing might proceed in such an example. These “double displacement” examples *do* cause conscious processing difficulty (Mazuka & Itoh, 1995).<sup>28</sup>

Gorrell (1995a,b) claims the contrast in processability between (15b) and (15c) can be derived via structural determinism (monotonicity of structural relations). However, we will see that, if we simply consider the structural relations before and after reanalysis, not only (15b) but also (15c) can be derived in a manner which preserves monotonicity.

Consider the schematic representation of a clause below:



<sup>28</sup>It should be noted that on-line experimental studies have only recently begun to investigate the Minimal Expulsion Strategy systematically. Hirose (1995) reports a series of studies investigating ditransitive structures which are slightly different from the examples discussed in the above paragraphs. These materials exhibited an ambiguity as to whether to displace only the nominative argument from the relative clause, leaving behind a dative and an accusative argument, or to displace both the nominative and dative arguments, leaving behind the accusative argument. An off-line preference task, asking subjects to choose the more natural of two sentences, actually showed a preference to expel the two arguments, but this was not replicated in an on-line experiment, which appeared to favour the minimal expulsion option instead. See also Yamashita (1994) for experimental evidence supporting the pre-verbal computation of a simplex ditransitive clause.

Let us say that the next word in the input is a noun. If the argument corresponding to  $Arg_1$  has to be relativised, then  $Arg_1$  will have to be displaced by an empty argument (or trace). If the argument corresponding to  $Arg_2$  has to be relativised, then both  $Arg_1$  and  $Arg_2$  have to be displaced. Let us consider the displacement of  $Arg_1$  first. An alternative way of looking at this, as Gorrell (1995a,b) has noted, is that everything *except*  $Arg_1$  is “lowered” (that is, in present terms, tree-lowering is performed on  $V'$ ), and a new S node is created (call it  $S_2$ ) which immediately dominates a newly created empty category in subject position. Then  $S_2$  is adjoined as a premodifier to the noun. The noun is postulated to head  $Arg_3$ , which is attached as a co-argument to  $Arg_1$ . This is illustrated in Fig. 5.

Now consider the displacement of both  $Arg_1$  and  $Arg_2$ . Gorrell (1995a,b) explains the difficulty of utterances requiring such double displacement in terms of the need to “delete” a domination relation between (in our terms)  $V'$  and  $Arg_2$ . In fact, however, this double displacement *can* be derived in an analogous manner to the single displacement example noted above. In this case, we lower the head node,  $V^0$ , and reconstruct a relative clause structure by adding the relevant nodes up to  $S_2$ , including *two* empty argument positions. This is illustrated in Fig. 6. As the reader can verify, this, as well as the single argument displacement, preserves informational monotonicity, since the original position of  $V^0$  dominates the post-reanalysis position.<sup>29</sup>

In contrast to Gorrell’s (1995a,b) model, where the distinction between single and double displacement is accounted for in terms of the parser’s inability to withdraw structural statements *at the initial point of disambiguation*, we would like to propose instead that the difficulty of examples such as (15c) may be due to the parser initially performing a mis-reanalysis, which only becomes apparent at a later point of processing, from where recovery is difficult.

The standard definition for tree-lowering offered above will obviously not suffice to deal with this type of example, since here we must add structure [including a new sentential node, and empty argument position(s)] which is not part of the subtree projection of the new head noun found in the input. The definition of tree-lowering has therefore been extended so that this extra structure can be built as part of the operation. The parser includes an *argument projection* operation, which, on the input of a head, checks for the

---

<sup>29</sup>In fact, the structure Gorrell proposes for this example includes an empty INFL node as a right sister of the VP. The double displacement example could be ruled out if we allow the verb to raise via verb-movement into this position. In this case, the last word to be processed will be in the INFL position, and thus the  $V^0$  node will no longer be accessible for lowering. However, this would rule out all reanalyses involving displacement of an object, but in some cases such examples are possible without conscious processing difficulty, as we shall see on pp. 483–484. Furthermore, there is very little convincing evidence for the existence of an INFL projection in Japanese clausal syntax (Fukui, 1986; see also Sells, 1995).

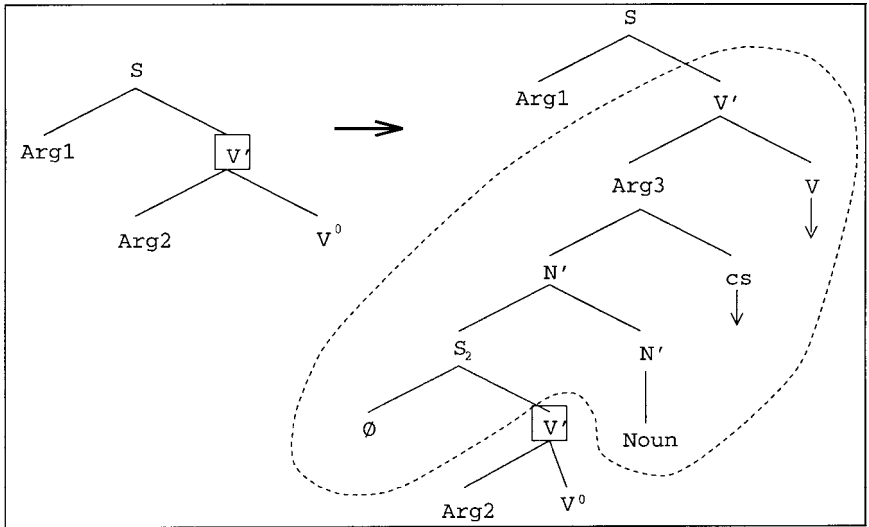


FIG. 5. Single displacement via the application of lowering at the  $V'$  node. The material enclosed within the dotted line is inserted into the structure on reanalysis. The position marked by "cs" represents the case marker expected in the input.

presence of the required arguments, and, in the case of a verbal head, adds empty categories for any arguments which are missing. It is this operation which is employed in the extended definition of tree-lowering. Where lowering is applied to a head-projection, *argument projection* is reapplied, so that, in cases where the arguments of a verb are displaced by reanalysis, the embedded clause structure is "regrown", including any necessary empty argument positions (see Appendix). In the examples we have been considering, the "regrown" embedded clause can then be attached as a relative clause to the incoming noun,<sup>30</sup> and this noun can then be attached as a co-argument to the displaced arguments (Sturt, 1994, contains details of this). Given the revised definition, either single or double displacement can be derived, depending on the node at which tree-lowering is applied. This means that, in order to account for the contrast in difficulty between (15c) and (15b), we will crucially have to appeal to the search strategy which the parser uses in finding a node for lowering. It will be clear that the bottom-up search we motivated in the previous section for English will predict exactly the opposite results for these Japanese examples. This is because the last word to be attached into a clause will be a verb and, therefore, at the point where the parser fails to attach the head noun, the lowest node accessible to

<sup>30</sup>In fact, it is *adjoined* rather than attached, since the relative clause *modifies* the head noun.

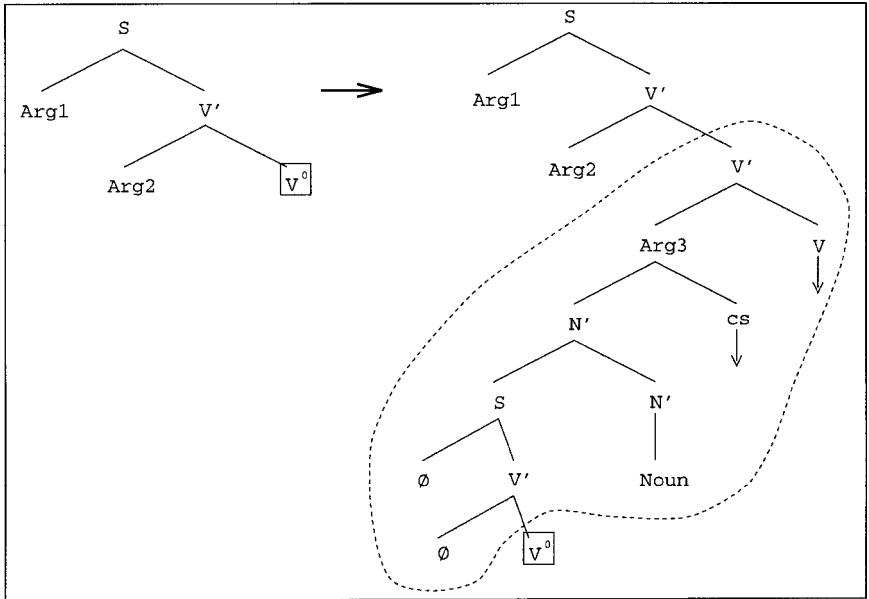


FIG. 6. Double displacement via the application of lowering at the  $V^0$  node.

the lowering operation will be the node immediately dominating this verb—that is, in the above schemata, the  $V^0$  node. This means that, if the parser begins its search of the accessible nodes at the bottom, the  $V^0$  node will be the first to be tried, and (given that the original clause contains two arguments) the embedded clause will be reconstructed with two empty arguments, in effect resulting in a double displacement. On the other hand, single displacement, which is known to be easy for Japanese perceivers, will be predicted to be *more difficult*, since it corresponds to choosing a lowering site which is higher in the structure.

If we reconsider the single displacement example (15b), repeated below as (16), we see how the bottom-up search strategy wrongly predicts a conscious garden path effect:

16. Yamasita ga yuuzin wo houmonsita siriai ni tegami wo kaita.  
 yamasita NOM friend ACC visited acquaintance DAT letter ACC wrote  
 “Yamasita wrote a letter to an acquaintance who visited his friend”.

Taking the  $V^0$  node immediately dominating *houmonsita* (“visited”) as the node chosen for lowering, the parser will displace both subject (*Yamasita ga*) and object (*yuuzin wo*) into the main clause.

This will result in an ungrammatical continuation, in which the verb *kaita* (“wrote”) takes two accusative arguments instead of one, and we must

wrongly predict a garden path effect when the parser notices this downstream (17):<sup>31</sup>

17. \*Yamasita ga yuuzin wo [ $\emptyset_{nom}$   $t_{acc}$  houmonsita] siriai<sub>i</sub> ni tegami  
 yamasita NOM friend ACC visited acquaintance DAT letter  
 wo kaita.  
 ACC wrote  
 (No grammatical translation)

## Top-down Search

A moment's reflection reveals that, if we want to reproduce the "minimal expulsion" effects using the tree-lowering operation for the type of examples discussed here, we should define a preference to perform lowering at as high a site as possible. It therefore seems reasonable to postulate a top-down search for Japanese.

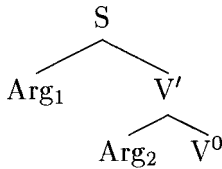
As we have almost no on-line experimental data concerning the processing of this type of example, the following discussion is necessarily speculative, and should be seen as one possible way in which a top-down search could proceed. It should be noted in particular that the parser may be sensitive to more types of information than the simply configurational issue of whether a node is high or low in a tree structure. One such factor may be the pragmatic plausibility of relativisation. Relativisation involves co-indexing a head noun with an argument position (which is occupied by a gap site in the relative clause). Assuming a model which allows a certain degree of interaction of non-syntactic knowledge in making parsing decisions (cf. Crain & Steedman, 1985), it may be that the processor takes into account the plausibility of establishing the referent of the head noun in the argument position concerned. Other factors which may play a part here include the valency preferences of the verb, and the obliqueness of the argument to be relativised. Another factor which has been shown to be important is the case-marking on the relativising head noun (Inoue, 1991).<sup>32</sup>

<sup>31</sup>In fact, this structure will violate the so-called *double-o constraint*, which bars the overt presence of two accusative marked NPs (in our terms, PPs) as arguments of the same predicate (see Kuroda, 1988, for details of this constraint).

<sup>32</sup>For example, native speakers seem to avoid on-line parsing decisions which would require postulating two nominative-marked NPs as arguments of the same predicate. This means that, in the class of examples which we have been discussing, if the relativising head noun is marked with nominative case, then there will be a preference against displacing the (nominative-marked) embedded subject to the matrix clause (i.e. a preference for the null-displacement option).



Imagine a two-argument clause has been built, resulting in the structure illustrated below, and that this is followed immediately by a noun, which must be incorporated somehow into the analysis:



One possible search strategy is that the parser considers each accessible node in a top-down order (S, V', V<sup>0</sup>), until a plausible relativisation site is found.

In a top-down search, the first node to be considered for lowering is S. This solution corresponds to retaining all arguments in the relative clause. We will call this *null displacement*. Since the relative clause must contain a gap, and all of the arguments are overt, the gap must represent an adjunct, which, as we have seen above, must be either temporal or locative. Thus, the null-displacement option will only be available if the semantic content of the relativising head noun is a plausible time or location for the semantic content of the relative clause to have taken place.

If this is not possible, we move on to consider node V', and attempt to lower. This option corresponds to relativising Arg<sub>1</sub>. A new sentential node will be created, dominating an empty argument in the position occupied by Arg<sub>1</sub>. This means that S will remain as the matrix sentential node, and will continue to immediately dominate Arg<sub>1</sub>, or, to put it another way, Arg<sub>1</sub> will be displaced from the relative clause. This will be possible if it is plausible to co-index the referent of the head noun with the empty element in the position of Arg<sub>1</sub>. If this is not possible, then the parser will descend to the next node, V<sup>0</sup>, and attempt to relativise Arg<sub>2</sub>. In the context shown above, this kind of search will predict the “minimal expulsion” strategy, with null displacement preferred.

Consider first the null-displacement example (15a), repeated below as (18):

18. [Mary<sub>i</sub> ga sinseihin wo <sub>t<sub>loci</sub></sub> kaihatu<sub>i</sub>sita] kaisya<sub>i</sub> ga tubereta.  
 Mary NOM new product ACC developed company NOM went bankrupt  
 “The company where Mary developed the new product went bankrupt”.

At the point when *kaisya* (company) is found in the input, the first node to be considered will be the top S node. The parser considers the relativisation corresponding to this node (i.e. adverbial relativisation), which is found to be plausible, since a company is a plausible location for Mary to have developed a new product. Thus no difficulty is predicted; indeed, this sentence does not cause conscious processing difficulty.

Now consider (15b) repeated again as (19):

19. Yamasita ga [ $t_{nom/i}$  yuuzin wo houmonsita] siriai<sub>i</sub> ni tegami wo  
 yamasita NOM friend ACC visited acquaintance DAT letter ACC  
 kaita.  
 wrote.  
 “Yamasita wrote a letter to an acquaintance who visited his friend”.

As before, the parser first builds the transitive clause with *houmonsita* (“visited”) as the main verb. This time, null displacement is not a possibility, since *siriai* (“acquaintance”) is not a plausible location or time. This means that the processor considers the next node down as a lowering site. This node will be the constituent covering the object and verb, *yuuzin wo houmonsita*. Accordingly, the subject argument, *yamasita ga*, is displaced, and a relative clause structure is built with an empty subject position. This analysis remains grammatical throughout the parse, and the structure will be unproblematic for the processor.

Finally, consider (15c) repeated below as (20):

20. Yamasita ga yuuzin wo [ $\emptyset_{nom} t_{acc/i}$  houmonsita] kaisya<sub>i</sub> de mikaketa.  
 Yamasita NOM friend ACC visited company LOC saw  
 “Yamasita saw his friend at the company he visited”.

Example (20) is complicated by the fact that the string is not only locally but also globally ambiguous. The other reading is one in which the main clause contains two empty arguments, and the initially built clause remains intact as an adjunct relative. The null context strongly disfavours this reading, in which two uncontrolled gaps appear in the matrix clause, but it is possible to create a prior context which provides discourse control for both of these arguments, as in the question in (21a) below. In this case, the utterance is considerably easier to process:

- 21a. anata wa doko no kaisya de Piitaa wo mikaketa no?  
 you TOP where GEN company LOC Peter ACC saw Q  
 “At which company did you see Peter?”
- b.  $\emptyset_{nom} \emptyset_{acc}$  [ $t_{loc/i}$  Yamasita ga yuuzin wo houmonsita] kaisya<sub>i</sub> de  
 Yamasita NOM friend ACC met company LOC saw  
 mikaketa.  
 “I saw him at the company where Yamasita visited his friend”.

At the point where *houmonsita* (“visited”) is attached, the parser will have built a simple transitive clause with both nominative and accusative arguments overt. On encountering the noun *kaisya*, the first option to be considered is the adjunct relativisation corresponding to null displacement. This analysis is not implausible, since a company is a reasonable location for *Yamasita* to meet his friend. Let us say that the parser initially adopts this analysis. At the point when the final verb *mikaketa* (“saw”) is encountered, neither of its nominative or accusative arguments is overtly present in the

main clause. On the null context, this means that there are two uncontrolled arguments. However, raising the subject and object from the lower clause rectifies this situation. The two empty arguments in the lower clause are now both controlled, the accusative argument, marked  $\emptyset_{acc}$ , is grammatically controlled by the head noun of the relative, *kaisyā*, and the nominative argument,  $\emptyset_{nom}$ , is pragmatically controlled by the matrix subject, *Yamasita ga*. The explanation of the difficulty is that this raising of the two arguments cannot be derived via tree-lowering. This is because, by the time the disambiguating final verb *mikaketa* is encountered in the input, the relevant node for lowering will no longer be accessible, since it will be embedded inside the relative clause.

## EASY DOUBLE DISPLACEMENTS

The present analysis predicts that double displacement should be possible if, at the initial point of reanalysis (i.e. where the immediately post-clausal noun is encountered), the need to perform lowering consistent with double displacement is obvious. This is in contrast to both Gorrell (1995a,b) and Weinberg (1993, 1995), both of whom propose models which allow the displacement of an overt subject but not an overt object.<sup>33</sup>

Mazuka and Itoh (1995) give the following example, which reportedly causes no conscious processing difficulty, despite the fact that both the subject and object have to be displaced:

22. Hirosi ga aidoru kasyu wo [ $\emptyset_{nom} t_{acc/i}$  kakusita] kamera<sub>i</sub> de totta.  
 Hirosi NOM popular singer ACC hid camera with photographed  
 "Hirosi photographed the popular singer with the camera he was hiding".

We assume, as before, that the overt nominative and accusative arguments are initially structured as arguments of the verb *kakusita* ("hid"). On encountering the head noun *kamera*, the parser first considers the null-displacement option, which is found to be implausible ("the camera where/when Hirosi hid the popular singer"). The single displacement option is similarly ruled out ("the camera which hid the popular singer"). Finally, the double displacement option is considered, and is found to be plausible ["the camera which (somebody) hid"]. This option is adopted, and the remaining processing proceeds without trouble.

A similar effect can be seen if we consider topicalisation. In Japanese, topicalised elements, which are given an overt morphological marker *wa*, almost invariably occur in the matrix clause,<sup>34</sup> though they may control a

<sup>33</sup>Though as we have seen above, if the lowering of a verb is permitted, Gorrell's model *will* allow the displacement of an overt object.

<sup>34</sup>Though see Kuroda (1988) for some limited exceptions.

“gap” at any level of embedding. Below we reproduce the double displacement example (15c) with the nominative marked argument topicalised. This is reported to be considerably easier to process than the non-topicalised version.

23. Yamasita wa yuuzin wo [ $\emptyset_{nom}$   $t_{acc}$  houmonsita] kaisya<sub>i</sub> de mikaketa.  
 Yamasita TOP friend ACC visited company LOC saw  
 “(as for) Yamasita (he) saw his friend at the company he visited”.

In the top-down search described above, we hypothesised that, on reaching the head noun, *kaisya* (“company”), the processor first attempts to form a relative clause consistent with null displacement, with a locative relativisation reading. However, in (23), the parser can eliminate this option immediately, since it would involve a topicalised phrase *Yamasita wa* appearing in a subordinate clause. The next option to be tried will be the single displacement option, in which the constituent covering *yuuzin wo houmonsita* (“visited the friend”) is lowered. However, this may be discounted on the grounds of plausibility, since “company” is not a plausible subject for “visited”. The parser is then left with no choice but to go along with the double displacement option, which eventually turns out to be correct. Thus (23) is correctly predicted to be easier than its non-topicalised counterpart in (15c). However, on the bottom-up search, there would be no difference predicted, since the  $V^0$  node will be the first node chosen as a prospective lowering site in both cases.

## EXPLAINING DIFFERENCES IN SEARCH STRATEGIES

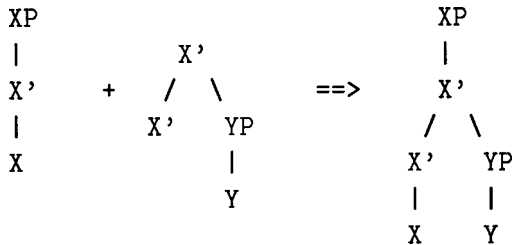
In the preceding sections, we have seen how the currently known data motivate a difference in search strategy for reanalysis in English and Japanese. In this section, we look in detail at the effect which reanalysis has on linguistic dependencies, and suggest that, rather than being explicitly parameterised in the parser, the difference in search strategies may be motivated by a strategy on the part of the parser to preserve as many dependencies as possible. It will be noted that this account shares features with Frazier’s (1994) Minimal Revisions Principle.

### Three Types of Lowering

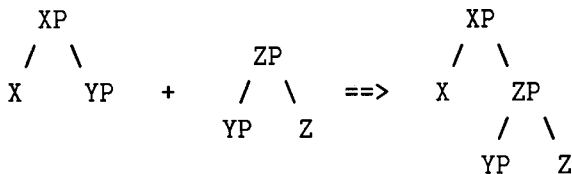
As discussed above, the use of informational monotonicity at the level of purely structural relations may result in non-monotonic behaviour at the level of “secondary” linguistic dependencies. In this section, we discuss three different scenarios in which lowering may apply, and in each case we discuss the consequences for the dependencies between head and satellites. Note that the lowering operation discussed in this paper is what we may call *retrospective* lowering; that is, where the disambiguating word is *preceded* by

the constituent that it lowers. Another possible operation, which we do not discuss here, is *anticipatory* lowering. In this case, the disambiguating word will precede the lowered constituent, where all the terminal nodes dominated by the lowered constituent are dangling nodes not yet dominating lexical material (e.g. an adjunction-like operation is performed on a predicted branch of the tree).

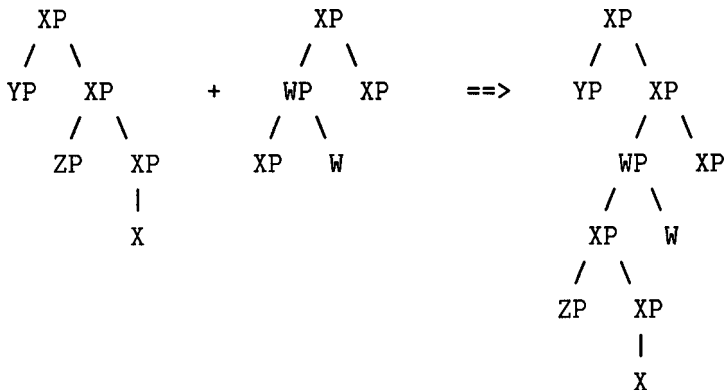
1. HEAD EXTENSION: a head projection is extended through the insertion of material at an intermediate point. *Example*: post-modifier attachment.



2. SATELLITE DETACHMENT: a satellite projection is broken. *Example*: *John knows the truth hurts*, where the satellite projection between the NP dominating *the truth* and the matrix VP is broken.



3. HEAD DETACHMENT: a head projection is split into two separate projections. *Examples* in discussion of Japanese data.



## Head-final and Head-initial Languages

Now let us consider the consequences of performing each of the three lowering operations in head-initial and head-final languages. We consider in particular the number of dependencies which must be broken in order to perform lowering on the above structures. By “dependency”, we mean a (licensing) relation between a satellite and its head *both of which have been encountered in the input so far*. In other words, we do not consider a dependency to exist between a head and its satellite if either of the two have not yet been encountered in the input.<sup>35</sup>

Let us consider *head extension* first. On standard X-bar assumptions, this corresponds to the attachment of a modifier. In a head-final language, in which only pre-modifiers (and no post-modifiers) exist, this operation can only be employed if the word which heads the projection to be extended has not yet been encountered in the input, and not via retrospective lowering as we have defined it. In a head-initial language like English, however, which allows post-modifiers, head-extension will be possible via retrospective lowering. In this case, the lowering operation will *add* one dependency between the head and the newly adjoined constituent.

Now consider *satellite detachment*. In a head-final language, all satellites precede the head. This means that, once the head has been attached, none of the satellites will be *accessible* in the sense defined above, since none will dominate the head (i.e. the last word to be incorporated). So in a head-final language, satellite detachment will only be possible in cases where the licensing head has not yet been reached in the input, and therefore will not break a dependency; that is, satellite detachment will be restricted to cases of *anticipatory lowering*. In a head-initial language, by contrast, any satellite which is preceded by its head will remain accessible to immediately following lexical material when the satellite phrase in question has been completed, and subsequent detachment of the satellite will result in breaking the dependency between that satellite and its head.

Finally, consider *head detachment*. This is very likely to be found in a head-final language. It corresponds to the case where a constituent, call it XP, has been completed, but the word subsequently found in the input (call it W), requires one of the nodes on X's head projection. This node is attached to W's left, and replaced with the root node of W's subtree projection. The word W, requiring a constituent on its left, may be a postposition, for example, in a head-final language. Head detachment may break any number of dependencies. In a head-final language, at the point where a constituent

---

<sup>35</sup>Note that, in modular theories such as GB, there may be a number of licensing relations between a single head and a single satellite. In this case, we do not count each single licensing relation as a single dependency, but treat the entire complex of licensing relations as one dependency.

has been built with a head and all its satellites, but no further attachments have been made, all nodes on the head projection of that constituent will be accessible. If a head-detaching lowering operation subsequently has to be performed, then we will assume that the processor attempts to lower at a node consistent with breaking the smallest number of dependencies possible, and this will coincide with Inoue's Minimal Expulsion Strategy.

It can be seen that the minimal expulsion preference can be derived via a preference to lower at the highest node possible, thus maintaining intact the largest number of dependencies between the head and its satellites. The reanalysis of an initially built clause as a relative in the examples we have been discussing above may be seen as instances of schema 3 on p. 485 (abstracting away from syntactic details), where the word W corresponds to the post-clausal head noun, and XP is the clausal (verbal) projection.

In English, on the other hand, head extension and satellite detachment will be employed far more often. Since, as far as the number of broken dependencies is concerned, nothing hinges on the choice of lowering site for either of these, the processor may be following a different strategy, which may include a preference to lower nodes which have been created as recently as possible (i.e. assuming a right-branching structure, to choose a low site). Considerations such as these may well underlie the differing search strategies between the two languages.

## OTHER COMPUTATIONAL APPROACHES TO HUMAN SYNTACTIC REANALYSIS

Suzanne Stevenson's (1993, 1994) competition-based model derives from the constraints of a connectionist architecture a number of very interesting predictions for syntactic processing. The model is similar to that proposed here in the sense that reanalysis is not seen as qualitatively different from simple attachment. In fact, Stevenson's constraint that reanalysis should only involve nodes on the right edge of the tree, which follows from the space constraints inherent in the connectionist architecture, leads to a reanalysis operation which is very similar to tree-lowering constrained by *accessibility*, as defined here. The decay of network activation predicts an empirically supported recency preference in reanalysis as well as attachment, which means that the "bottom-up" search strategy which we have discussed here does not have to be stipulated. However, it is not clear how the model would perform in processing head-final languages, where, assuming an incremental processing regime, the recency preference would presumably predict the opposite of the "top-down", dependency-preserving search strategy motivated for Japanese in this paper. That is, a strategy which takes into account more grammatical information may be required in addition to the simple recency preference which is predicted in a decay of activation model.

Also, the constraints of Stevenson's model may in fact be too strong. For example, phrase structure cannot be postulated without explicit evidence from the input, so that, for example, in a sentence such as *Mary thinks John likes oranges*, processing difficulty is predicted on the input of *John*, because *thinks* does not select for an NP, and therefore no attachment site can be postulated from explicit evidence in the input up to that point in processing. On the other hand, in a model such as we have been discussing in this paper, which does not forbid non-lexically driven prediction, this problem need not arise, as we have seen. This strict constraint also makes it difficult to extend Stevenson's model to include more complex reanalysis operations, such as the extended version of tree-lowering described on pp. 477–480.

Lewis (1993) presents a comprehension model, NL-SOAR, which incorporates a syntactic reanalysis component. If, on the attachment of an incoming word, an inconsistency is detected within the local maximal projection to which the incoming word is attached, then NL-SOAR's "snip" operator can break a previous attachment within this maximal projection and reattach it elsewhere in the tree. This operation is more powerful than tree-lowering in the sense that the phrase detached by the snip operator does not have to be reattached in a position which is dominated by the projection of the incoming, disambiguating word. This results in an impressive range of correct predictions. For example, in a sentence such as *Is the block on the table red?*, the disambiguating word *red* can trigger the snip operator to detach the PP *on the table*, which has previously been attached as the complement of the copula, and reattach it as an adjunct of *block*, correctly predicting this sentence to cause no processing difficulty; however, lowering will not account for this, since the post-reanalysis position of *on the table* is not dominated by the projection of the disambiguating word *red*. However, NL-SOAR does overgenerate<sup>36</sup> on a class of examples such as: "*The psychologist told the woman that he was having trouble with to leave*" and "*The boy put the book on the table on the shelf*", which both involve reattaching material into a preceding phrase, which is not possible in Gorrell's model, and also cannot be generated via tree-lowering.

## CONCLUSIONS

Our aim has been to show how we can retain the intuitive appeal of a D-theory based approach, while also providing an explanation of the limitations on the human parser's ability to reanalyse structure in terms of a constrained search space.

One of the hallmarks of the D-theory approach is that there is no major conceptual distinction between (unconscious) reanalysis and attachment.

---

<sup>36</sup>By "overgenerate", we mean here that the parser is able to process examples which are actually conscious garden paths.



The definition of a *reanalysis* oriented attachment operation (i.e. tree-lowering) has led us to explore the possibility that there are “reanalysis ambiguities” just as there are attachment ambiguities, and this demands the consideration of preferences for the application of the tree-lowering operation, which have to be defined over and above the basic D-theoretic machinery. We have seen that the preference to “reanalyse low” in English has to be replaced by a preference to “reanalyse high” in Japanese, and we have speculated that this may be due to a “dependency-preserving” strategy of the parser. Thus, one research question which we intend to address in the future is that of whether the observed behaviour of the parser is a direct consequence of the attempt to preserve as many dependencies as possible, or whether it is simply the result of a compiled strategy which is known to preserve the maximum number of dependencies in the general case (but might not in particular cases). As a second avenue of research, we intend to give the model extra flexibility by defining a new underspecification formalism. On pp. 472–473, we examined cases in which the model is unable to reanalyse where humans do not encounter conscious difficulty. A new area of research, which we are currently exploring (Sturt & Crocker, in press), consists in revising the underspecification formalism in such a way that it does not rely so heavily on the purely configurational notions of classical D-theory, thereby allowing the parser more freedom to reanalyse.

Manuscript received June 1995

Revised manuscript received April 1996

## REFERENCES

- Abney, S.P. (1987). Licensing and parsing. In *Proceedings of NELS, 17*, 1–15. Amherst, MA: University of Massachusetts.
- Abney, S.P. (1989). A computational model of human parsing. *Journal of Psycholinguistic Research, 18*, 129–144.
- Adams, B.C. (1995). *A model for strategic reanalysis in sentence processing*. Unpublished manuscript, University of Virginia.
- Barton, E., & Berwick, R. (1985). Parsing with assertion sets and informational monotonicity. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 769–771. Los Angeles, CA, August.
- Cornell, T.L. (1994). On determining the consistency of partial descriptions of trees. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 163–170. La Cruces, NM, June.
- Crain, S., & Steedman, M. (1985). On not being led up the garden path: The use of context by the psychological parser. In D. Dowty, L. Karttunen, & A. Zwicky (Eds), *Natural language processing: Psychological, computational and theoretical perspectives*, pp. 320–358. Cambridge: Cambridge University Press.
- Crocker, M.W. (1992). *A logical model of competence and performance in the human sentence processor*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh.
- Crocker, M.W. (1994). On the nature of the principle-based sentence processor. In C. Clifton, L. Frazier, & K. Rayner (Eds), *Perspectives on sentence processing*, pp. 245–266. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

- 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.
- Fodor, J.D., & Inoue, A. (1994). The diagnosis and cure of garden paths. In V. Teller (Ed.), special issue of *Journal of Psycholinguistic Research*, 23, 405–432.
- Frazier, L. (1978). *On comprehending sentences: Syntactic parsing strategies*. PhD thesis, University of Connecticut, Storrs, CT.
- Frazier, L. (1987). Syntactic processing: Evidence from Dutch. *Natural Language and Linguistic Theory*, 5, 519–559.
- Frazier, L. (1994). Sentence (re)-analysis. Paper presented at the 7th Annual CUNY Conference on Human Sentence Processing, New York, March.
- Frazier, L., & Clifton, C. (1995). *Construal*. Cambridge, MA: MIT Press.
- Frazier, L., & Rayner, K. (1987). Resolution of syntactic category ambiguities: Eye movements in parsing lexically ambiguous sentences. *Journal of Memory and Language*, 26, 505–526.
- Fukui, N. (1986). *A theory of category projection and its applications*. Unpublished PhD dissertation, MIT, Cambridge, MA.
- Gibson, E., Pearlmutter, N., Canesco-Gonzalez, E., & Hickok, G. (1996). Recency preference in the human sentence processing mechanism. *Cognition*, 59, 23–59.
- Gorrell, P. (1995a). Japanese trees and the garden path. In R. Mazuka & N. Nagai (Eds), *Japanese sentence processing*, pp. 331–350. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Gorrell, P. (1995b). *Syntax and parsing*. Cambridge: Cambridge University Press.
- Hirose, Y. (1995). Garden paths and the minimal everything principle in Japanese relative clauses. Poster presented at the 8th Annual CUNY Conference on Human Sentence Processing, Tucson, AZ, March.
- Inoue, A. (1991). *A comparative study of parsing in English and Japanese*. PhD thesis, University of Connecticut, Storrs, CT.
- Inoue, A., & Fodor, J.D. (1995). Information-paced parsing of Japanese. In R. Mazuka & N. Nagai (Eds), *Japanese sentence processing*, pp. 9–63. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Joshi, A.K., Levy, L.S., & Takahashi, M. (1975). Tree adjunct grammars. *Journal of Computer and System Sciences*, 10, 136–163.
- Joshi, A.K., Vijay-Shanker, K., & Weir, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In P. Sells, S.M. Shieber, & T. Wasow (Eds), *Foundational issues in natural language processing*, pp. 31–81. Cambridge, MA: MIT Press.
- Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, 2, 15–47.
- Kuroda, S.-Y. (1988). Whether we agree or not: A comparative syntax of English and Japanese. In W.J. Poser (Ed.), *Papers from the second workshop on Japanese syntax*, pp. 103–143. Stanford, CA: CSLI Publications.
- Lewis, R.L. (1993). *An architecturally-based theory of human sentence comprehension*. PhD thesis, Carnegie-Mellon University, Pittsburgh, PA.
- Marcus, M., Hindle, D., & Fleck, M. (1983). D-theory: Talking about talking about trees. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pp. 129–136. Cambridge, MA, June.
- Mazuka, R., & Itoh, K. (1995). Can Japanese speakers be led down the garden path? In R. Mazuka & N. Nagai (Eds), *Japanese sentence processing*, pp. 295–329. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Mazuka, R., & Nagai, N. (Eds) (1995). *Japanese sentence processing*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Mazuka, R., Itoh, K., Kiritani, S., Niwa, S., Ikejiri, K., & Naitoh, K. (1989). Processing of Japanese garden-path, center-embedded, and multiply-left-embedded sentences: Reading

- time data from an eye movement study. *Annual Bulletin of the Research Institute of Logopedics and Phoniatrics*, 23, 187–212.
- Milward, D. (1994). Dynamic Dependency Grammar. *Linguistics and Philosophy*, 17, 561–605.
- Milward, D. (1995). Incremental interpretation of categorial grammar. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 119–126. University College, Dublin, March.
- Mitchell, D.C. (1989). Verb guidance and other lexical effects in parsing. *Language and Cognitive Processes*, 4, 123–154.
- Partee, B., ter Meulen, A., & Wall, R.E. (1993). *Mathematical methods in linguistics*. Dordrecht: Kluwer Academic.
- Phillips, C. (1995). Right association in parsing and grammar. In C. Schütze, J. Ganger, & K. Broihier (Eds), *Papers on Language Processing and Acquisition*, pp. 37–93. MIT Working Papers in Linguistics Vol. 26. Cambridge, MA: MIT Press.
- Pritchett, B.L. (1992). *Grammatical competence and parsing performance*. Chicago, IL: University of Chicago Press.
- Pulman, S.G. (1986). Grammars, parsers and memory limitations. *Language and Cognitive Processes*, 1, 197–225.
- 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 Behavior*, 22, 358–374.
- Schabes, Y., Abeillé, A., & Joshi, A.K. (1988). New parsing strategies for tree adjoining grammars. In *Proceedings of the 12th International Conference in Computational Linguistics*, pp. 578–583. Budapest, August.
- Sells, P. (1995). Korean and Japanese morphology from a lexical perspective. *Linguistic Inquiry*, 26, 277–325.
- Shieber, S., & Johnson, M. (1993). Variations on incremental interpretation. *Journal of Psycholinguistic Research*, 22, 287–318.
- Stabler, E.P. (1994a). *Parsing for incremental interpretation*. Unpublished manuscript, University of California, Los Angeles, CA.
- Stabler, E.P. (1994b). *Syntactic preferences in parsing for incremental interpretation*. Unpublished manuscript, University of California, Los Angeles, CA.
- Steedman, M.J. (1989). Grammar, interpretation and processing from the lexicon. In W. Marslen-Wilson (Ed.), *Lexical representation and process*, pp. 463–504. Cambridge, MA: MIT Press.
- Stevenson, S. (1993). A competition-based explanation of syntactic attachment preferences and garden path phenomena. In *Proceedings of the 31st Meeting of the Association for Computational Linguistics*, pp. 266–273. Columbus, OH, June.
- Stevenson, S. (1994). Competition and recency in a hybrid network model of syntactic disambiguation. *Journal of Psycholinguistic Research*, 23, 295–321.
- Sturt, P. (1994). *Unconscious reanalysis in an incremental processing model*. Unpublished manuscript, Centre for Cognitive Science, University of Edinburgh.
- Sturt, P., & Crocker, M.W. (1995). Incrementality and monotonicity in syntactic parsing. In D. Milward & P. Sturt (Eds), *Incremental interpretation*, pp. 23–66. Edinburgh Working Papers in Cognitive Science, Vol. 11. Edinburgh: Centre for Cognitive Science.
- Sturt, P., & Crocker, M.W. (in press). Thematic monotonicity. *Journal of Psycholinguistic Research*.
- Thompson, H., Dixon, M., & Lamping, J. (1991). Compose–reduce parsing. In *Proceedings of the 29th Meeting of the Association for Computational Linguistics*, pp. 87–97. Berkeley, CA, June.

- Vijay-Shanker, K. (1992). Using descriptions of trees in a tree adjoining grammar. *Computational Linguistics*, 18, 481–517.
- Weinberg, A. (1993). Parameters in the theory of sentence processing: Minimal Commitment Theory goes East. *Journal of Psycholinguistic Research*, 22, 339–364.
- Weinberg, A. (1995). Licensing constraints and the theory of language processing. In R. Mazuka & N. Nagai (Eds), *Japanese sentence processing*, pp. 235–255. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Yamashita, H. (1994). *Processing of Japanese and Korean*. PhD thesis, Ohio State University, Columbus, OH.

## APPENDIX

The purpose of this Appendix is to give a more precise characterisation of the implementation by giving a brief description of the basic algorithm and data structures employed.

### Grammar representation

Each word in the lexicon is associated with an argument frame, and a lexical category (e.g. verb, noun). To each lexical category corresponds a *grammar entry*, which is a quintuple  $\langle C, D, R, Left, Right \rangle$ , where  $C$  is the name of the lexical category (e.g.  $V^0$ ,  $N^0$ , etc.),  $D$  is the set of dominance and precedence relations which describe the lexical subtree anchored in  $C$ ,  $R$  is the root node of the lexical subtree, and  $Left$  and  $Right$  are lists of the nodes in the subtree which are attachment sites for the category  $C$ , where the nodes in  $Left$  precede  $C$  and the nodes in  $Right$  follow  $C$ .

The grammar entry for the verb category in English (see p. 460) is as follows:

$$C = V^0$$

$$D = \{ \text{dom}(VP, V^0), \text{dom}(V^0, \text{Word}), \text{dom}(S, VP), \text{dom}(S, NP), \text{prec}(NP, VP) \}$$

$$R = S$$

$$Left = \langle NP \rangle$$

$$Right = \langle \rangle$$

Note that the grammar entry only represents properties of the general lexical category in question. Thus, the grammar entry for verbs, shown above, has an attachment site for a subject, reflecting the fact that all verbs in English must have a subject, but does not include attachment sites for the internal arguments of particular verbs. These attachment sites are projected with reference to the argument frames of each particular verb in the lexicon, when the verb is used.

### Global state

The global state of the parser is a triple  $\langle D, Root, Right \rangle$ , where  $D$  is the set of dominance and precedence relations which describe the global tree built up so far,  $Root$  is the root node of the global tree built up so far, and  $Right$  is the list of unsaturated attachment sites, which will have to be satisfied via *right attachment*. (Note that, by definition, there cannot be unsaturated left attachment sites in the global tree, since these will be inaccessible according to the theory of trees.)

The attachment of a new word is achieved by equating the relevant nodes, as required by the definitions of attachment and tree-lowering, adding the new structural relations from the word's grammar entry, projecting any necessary obligatory arguments and, if necessary, updating the global state so that it represents the relevant new right attachment sites, and new root node.

### Argument projection

When the parser incorporates a new word into the description, the global state is updated with reference to the new word's argument frame. Recall the grammar entry for English verbs given above. If an obligatorily transitive verb is processed, the resultant global state will include a new

NP node (call it  $NP_2$ ) in the list of right attachment sites. Also, the following relations will be added to the global description to encode a “dangling node”.

$$\{\text{dom}(\text{VP}, \text{NP}_2), \text{prec}(\text{V}^0, \text{NP}_2)\}$$

In the case of Japanese, which allows empty arguments, any arguments “missing” from the left of the new word will be added as empty arguments and, if necessary, the root node will be updated. For example, if a single accusative NP is followed by a ditransitive verb in the input, then the “missing” dative and nominative arguments will be added to the global description as empty categories, and the verbal projection extended to the S node dominating the subject NP.

### Non-lexical structure building

The grammar representation used here is not entirely lexicalised; that is, there are grammatical objects manipulated by the parser which do not have a lexical anchor. This is necessary to deal with Japanese relative clauses, which have no overt relative pronouns, or other explicit lexical signals to show that they can combine with following nouns. In the current implementation, we allow a clause containing at least one empty category to be “extended” to become a noun modifier; basically, the clause  $[_S \dots]$  is converted to  $[_N[_S \dots] [_N']]$  by adding the appropriate relations, and updating the root category as appropriate. A similar strategy would be needed to deal with English reduced relatives, for example.

As mentioned on pp. 467–468, we also allow a projected clausal complement in English to be extended down to the subject NP of the complement clause, even when the input lacks an explicit complementiser. This is also a case of building structure not directly justified by lexical input.

### Basic algorithm

The following is a pseudocode description of the basic control structure of the algorithm.

1. ● **If** Input is empty,
  - **Then** succeed.
  - **Else** go to 2.
2. **Project word**
  - (a) Read next word  $W$ .
  - (b) Find category  $C$  of  $W$ .
  - (c) Find grammar entry  $GE$  for  $C$ .
  - (d) Go to 3.
3. **Attachment**
  - (a) ● **If** the Global State is undefined (i.e.  $W$  is the first word of the input), and  $W$  does not require overt arguments to its left,
    - **Then** instantiate  $GE$  as the new Global Description, apply *argument projection*, and go to 2.
    - **Else**, go to 3b.
  - (b) ● **If** the preconditions for *right attachment* are met,
    - **Then** combine  $GE$  with the current Global Description via *right attachment*, perform *argument projection*, and go to 2.
    - **Else** go to 3c.
  - (c) ● **If** the preconditions for *left attachment* are met,
    - **Then** combine  $GE$  with the current Global Description via *left attachment*, perform *argument projection*, and go to 2.
    - **Else**, go to 4.

#### 4. Tree-lowering

- (a) Find the current node path  $P$ , i.e. the set of accessible nodes, ordered according to the ordering function  $F$  (see below for explanation).
- (b)
  - **If**  $P$  is empty,
  - **Then** fail.
  - **Else**, go to 4c.
- (c) Remove  $N$ , first node of  $P$ , leaving  $Rest$ , the remainder of  $P$ .
  - **If** the preconditions for *tree-lowering* are met at  $N$ ,
  - **Then** apply *tree-lowering* at  $N$ , and perform *argument projection*.
  - **Else**, set  $Rest = P$ , and go to 4b.

The different search strategies for the application of tree-lowering in English and Japanese are captured by setting different values to the ordering function  $F$ , mentioned in 4a. To obtain the bottom-up search used for English,  $F$  orders the relevant nodes from bottom to top, while the top-down search used for Japanese orders them from top to bottom. Though this difference in search strategies is presented as a stipulation in the actual implementation, we strongly suspect that it is actually the result of some dependency preserving strategy, as outlined on pp. 484–487.

It should also be noted that the algorithm, as presented above, is slightly simplified. In fact, the current implementation allows the Global Tree Description to be extended via non-lexical structure building, as briefly described in the previous subsection, before applying the composition operations. Furthermore, in cases where tree-lowering is applied on a head projection node (i.e. where head detachment is applied; see p. 485), argument projection is re-applied to the head word of the detached projection, in order to “regrow” empty arguments that will replace those “displaced” by the operation. This, as well as non-lexical structure-building, allows us to capture the rather complex version of tree-lowering described on pp. 477–480.

It should be noted that the serial character of the algorithm results in a preference for right over left attachment (since right attachment is tried first). We do not have any strong commitment to this preference, and we have not found any phenomena which exhibit a right/left attachment ambiguity, for which the preference would make crucial predictions.<sup>37</sup>

As a final remark, the reader may have noted that, because the parsing operations manipulate nodes and all their descendants as intact units, the parser might just as well have been described in terms of manipulating subtrees, without evoking descriptions, and indeed, for the purposes of clarity of exposition, we have often described aspects of our model in such terms in this paper.<sup>38</sup> However, future extensions to the underspecification formalism, mentioned in the Conclusions section, will almost certainly result in parsing operations which no longer have this property. In such a case, the use of descriptions will give the model valuable extra flexibility.

---

<sup>37</sup>Abney (1989) also reported that he could find no phenomena which exhibit the attach/attach-L conflict in his licensing-based model.

<sup>38</sup>This observation was brought to our attention by an anonymous reviewer.