

A Competition-Based Explanation of Syntactic Attachment Preferences and Garden Path Phenomena

Suzanne Stevenson

Department of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4 Canada
suzanne@cs.toronto.edu

Abstract

This paper presents a massively parallel parser that predicts critical attachment behaviors of the human sentence processor, without the use of explicit preference heuristics or revision strategies. The processing of a syntactic ambiguity is modeled as an active, distributed competition among the potential attachments for a phrase. Computationally motivated constraints on the competitive mechanism provide a principled and uniform account of a range of human attachment preferences and garden path phenomena.

1 A Competition-Based Parser

A model of the human parser must explain, among other factors, the following two aspects of the processing of a syntactic ambiguity: the initial attachment preferences that people exhibit, and their ability or inability to later revise an incorrect attachment. This paper presents a competition-based parser, CAPERS, that predicts critical attachment behaviors of the human sentence processor, without the use of explicit preference heuristics or revision strategies. CAPERS is a massively parallel network of processing nodes that represent syntactic phrases and their attachments within a parse tree. A syntactic ambiguity leads to a network of alternative attachments that compete in parallel for numeric activation; an attachment wins over its competitors when it amasses activation above a certain threshold. The competition among attachments is achieved solely through a technique called competition-based spreading activation (CBSA) (Reggia 87). The effective use of CBSA requires restrictions on the syntactic attachments that are allowed to compete simultaneously. Ensuring these network restrictions necessitates the further constraint that a stable state of the network can only represent a single valid parse state. The resulting network structure defines a limited set of com-

peting attachments that simultaneously define the initial attachments for the current input phrase, along with the reanalysis possibilities for phrases previously structured within the parse tree.

The competitive mechanism and its ensuing restrictions have profound consequences for the modeling of the human sentence processor. Whereas other models must impose explicit conditions on the parser's attachment behavior (Abney 89; Gibson 91; McRoy & Hirst 90; Pritchett 88), in CAPERS both initial attachment preferences and reanalyzability are a side effect of independently motivated computational assumptions. Furthermore, parsing models generally employ two different computational mechanisms in determining syntactic attachments: a general parser to establish the attachment possibilities, and additional strategies for choosing among them (Abney 89; Frazier 78; Gibson 91; McRoy & Hirst 90; Shieber 83). By contrast, CAPERS provides a more restrictive account, in which a single competitive mechanism imposes constraints on the parser that determine the potential attachments, as well as choosing the preferred attachment from among those.

The competitive mechanism of CAPERS also leads to an advantageous integration of serialism and parallelism. In order to conform to human memory limitations, other parallel models must be augmented with a scheme for reducing the number of structures that are maintained (Gibson 91; Gorrell 87). Such pruning schemes are unnecessary in CAPERS, since inherent properties of the competitive mechanism lead to a restriction to maintain a single parse state. However, in spite of this serial aspect, CAPERS is not a simple serial model. The network incorporates each input phrase through a parallel atomic operation that determines both the initial attachment for the current phrase and any revision of earlier attachments. Thus, CAPERS avoids the problems of purely serial or race-based models that rely on backtracking, which is cognitively implausible, or explicit revision strate-

gies, which can be unrestrictive (Abney 89; Frazier 78; Inoue & Fodor 92; McRoy & Hirst 90; Pritchett 88).

Other work (Stevenson 93b, 90) describes the detailed motivation for the CAPERS model, its explanation of serial and parallel effects in human parsing, and its predictions of a broad range of human attachment preferences. This paper focuses on the competitive mechanism described above. Section 2 briefly describes the implementation of the parser.¹ Section 3 discusses the constraints on the network structure, and Section 4 demonstrates the consequences of these constraints for the processing of attachment ambiguities. Section 5 summarizes how the competitive mechanism provides a principled and uniform account of the example human attachment preferences and garden path phenomena.

2 The Parsing Network

CAPERS dynamically creates the parsing network by allocating processing nodes in response to the input. Control of the parse is distributed among these nodes, which make attachment decisions solely on the basis of the local communication of simple symbolic features and numeric activation. The symbolic information determines the grammaticality of potential attachments, while numeric activation weighs the relative strengths of the valid alternatives. The spreading activation process allows the network to gradually settle on a set of winning attachments that form a globally consistent parse tree.

Building the Network

When an input token is read, the parser activates a set of phrasal nodes, or p-nodes, from a pool of \bar{X} templates; their symbolic features are initialized based on the input token's lexical entry. Figure 1 shows a sample \bar{X} template and its instantiation. Syntactic phrases are only allocated in response to explicit evidence in the input; top-down hypothesizing of phrases is disallowed because it greatly increases the complexity of the network. Next, the parser allocates processing nodes to represent the potential attachments between the current input phrase and the existing parse tree. Attachment nodes, or a-nodes, are established between potential *sisters* in the parse tree; each a-node connects to exactly two p-nodes, as shown in Figure 2. (In all figures, a-nodes are shown as squares, which are black when the a-node is fully activated.) Once the current phrase is connected to the existing network, each processing node iteratively updates its

¹CAPERS is implemented in Common Lisp, serially simulating the parallel processing of the network.

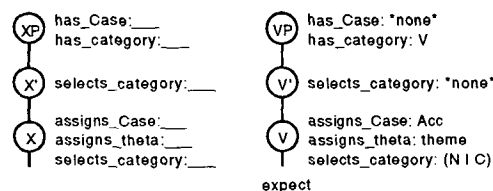


Figure 1: An \bar{X} template and sample instantiation.

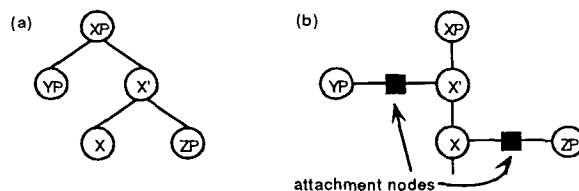


Figure 2: (a) The basic configuration of a phrase in \bar{X} theory. (b) Representation of these attachments as sister relations in CAPERS.

symbolic features and numeric activation, and outputs them to its neighbors. This network processing loop continues until the activation level of each a-node is either above a certain threshold θ , or is zero.² The set of active a-nodes in this stable state represents the current parse tree structure. At this point, the next input token is read and the process is repeated.

Grammaticality of Attachments

Unlike other connectionist parsers (Cottrell 89; Fianty 85; Selman & Hirst 85), CAPERS is a hybrid model whose limited symbolic processing abilities support the direct representation of the grammar of a current linguistic theory. In Government-Binding theory (GB) (Chomsky 81, 86; Rizzi 90), the validity of syntactic structures is achieved by locally satisfying the grammatical constraints among neighboring syntactic phrases. CAPERS directly encodes this formulation of linguistic knowledge as a set of simultaneous local constraints. Symbolic features are simple attribute/value pairs, with the attributes corresponding to grammatical entities such as Case and theta roles. The values that these attributes can assume are taken from a pre-defined list of atoms. GB constraints are implemented as equality tests on the values of certain attributes. For example, the Case Filter in GB states that every NP argument must receive Case. In CAPERS, this is stated as a condition that the attribute Case must receive a value when the attribute IsArgument equals True.

An a-node receives symbolic features from its p-

²The network always stabilizes in less than 100 iterations.

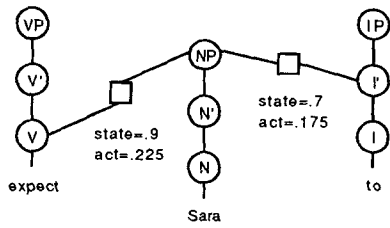


Figure 3: The NP can attach as a sister to the V or the I'. The attachment to the V has a higher grammatical state value, and thus a higher initial activation level.

nodes, which are used to determine the grammaticality of the attachment. If an a-node receives incompatible features from its two p-nodes, then it is an invalid attachment and it becomes inactive. Otherwise, it tests the equality conditions that were developed to encode the following subset of GB constraints: the Theta Criterion, the Case Filter, categorial selection, and the binding of traces. The algorithm outputs a numeric representation of the degree to which these grammatical constraints are satisfied; this state value is used in determining the a-node's activation level.

Choosing Preferred Attachments

Multiple grammatical attachments may exist for a phrase, as in Figure 3. The network's task is to focus activation onto a subset of the grammatical attachments that form a consistent parse tree for the input processed thus far. Attachment alternatives must be made to effectively compete with each other for numeric activation, in order to ensure that some a-nodes become highly activated and others have their activation suppressed. There are two techniques for producing competitive behavior in a connectionist network. The traditional method is to insert inhibitory links between pairs of competing nodes. Competition-based spreading activation (CBSA) is a newer technique that achieves competitive behavior indirectly: competing nodes vie for output from a common neighbor, which allocates its activation between the competitors. In a CBSA function, the output of a node is based on the activation levels of its neighbors, as in equation 1.

$$o_{ji} = \frac{a_j}{\sum_k a_k} \cdot a_i \quad (1)$$

where:

- o_{ji} is the output from node n_i to node n_j ;
- a_i is the activation of node n_i ;
- k ranges over all nodes connected to node n_i .

For reasons of space efficiency, flexibility, and cogni-

tive plausibility (Reggia et al. 88), CBSA was adopted as the means for producing competitive behavior among the a-nodes in CAPERS. Each p-node uses a CBSA function to allocate output activation among its a-nodes, proportional to their current activation level. For example, the NP node in Figure 3 will send more of its output to the attachment to the V node than to the I' node. The CBSA function is designed so that in a stable state of the network, each p-node activates a number of a-nodes in accordance with its grammatical properties. Since every XP must have a parent in the parse tree, all XP nodes must activate exactly one a-node. An X or X' node must activate a number of a-nodes equal to the number of complements or specifiers, respectively, that it licenses. The a-nodes enforce consistency among the p-nodes' individual attachment decisions: each a-node numerically ANDs together the input from its two p-nodes to ensure that they agree to activate the attachment.

A p-node that has obligatory attachments must at all times activate the appropriate number of a-nodes in order for the network to stabilize. However, since the phrase(s) that the p-node will attach to may occur later in the input, the parser needs a way to represent a "null" attachment to act as a placeholder for the p-node's eventual sister(s). For this purpose, the model uses processing nodes called phi-nodes to represent a "dummy" phrase in the tree.³ Every X and X' node has an a-node that connects to a phi-node, allowing the possibility of a null attachment. A phi-node communicates default symbolic information to its a-node, with two side effects. The a-node is always grammatically valid, and therefore represents a default attachment for the p-node it connects to. But, the default information does not fully satisfy the grammatical constraints of the a-node, thereby lowering its activation level and making it a less preferred attachment alternative.

3 Restrictions on the Network

The competitive mechanism presented thus far is incomplete. If all possible attachments are established between the current phrase and the existing network, CBSA cannot ensure that the set of active a-nodes forms a consistent parse tree. CBSA can weed out locally incompatible a-nodes by requiring that each p-node activate the grammatically appropriate number of a-nodes, but it cannot rule out the simultaneous activation of certain incompatible attachments that are farther apart in the tree. Figure 4 shows the types of structures in which CBSA is an insufficient

³Phi-nodes also represent the traces of displaced phrases in the parse tree; see (Stevenson 93a, 93b).

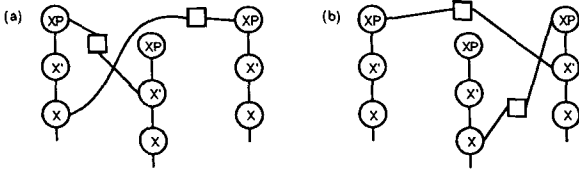


Figure 4: Example pairs of incompatible attachments that CBSA alone cannot prevent from being active simultaneously.

competitive mechanism. Both cases involve violations of the proper nesting structure of a parse tree. Since CBSA cannot rule out these invalid structures, the parsing network must be restricted to prevent these attachment configurations. The parser could insert inhibitory links between all pairs of incompatible a-nodes, but this increases the complexity of the network dramatically. The decision was made to instead *reduce* the size and connectedness of the network, simultaneously solving the tree structuring problems, by only allowing attachments between the current phrase and the right edge of the existing parse tree.

Limiting the attachment of the current phrase to the right edge of the parse tree rules out all of the problematic cases represented by Figure 4(a). Interestingly, the restriction leads to a solution for the cases of Figure 4(b) as well. Since there is no global controller, each syntactic phrase that is activated must be connected to the existing network so that it can participate in the parse. However, sometimes a phrase cannot attach to the existing parse tree; for example, a subject in English attaches to an inflection phrase (IP) that follows it. The network connections between these unattached phrases must be maintained as a stack; this ensures that the current phrase can only establish attachments to the right edge of an immediately preceding subtree. The stack mechanism in CAPERS is implemented as shown in Figure 5: a phrase pushes itself onto the stack when its XP node activates an a-node between it and a specially designated stack node. Because the stack cannot satisfy grammatical constraints, stack node attachments are only available for the XP. The flexibility of CBSA allows the stack to activate more than one a-node, so that multiple phrases can be pushed onto it. The surprising result is that, by having the stack establish a-nodes that compete for activation like normal attachments, the indirect competitive relationships within the network effectively suppress all inconsistent attachment possibilities, including those of Figure 4(b).

This result relies on the fact that any incompatible a-nodes that are created either directly or indirectly

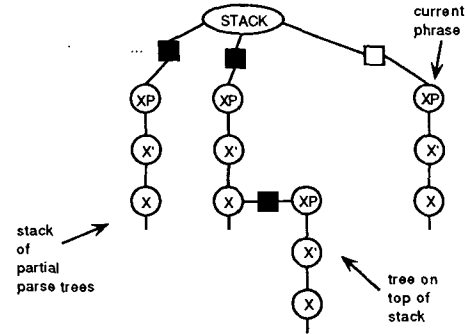


Figure 5: The stack is implemented as a degenerate p-node that can activate attachments to XP nodes.

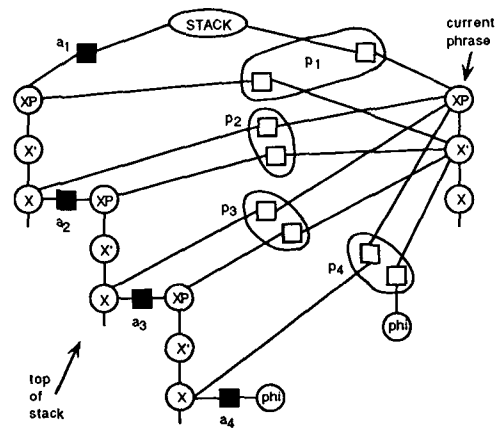


Figure 6: Attachments a_1 – a_4 were previously activated. To attach the current phrase to the tree on the stack, the following must occur: exactly one of the prior attachments, a_i , must become inactive, and the corresponding pair of attachments, p_i , must become active. This relationship holds for a tree of arbitrary depth on the stack.

compete with each other through CBSA. To guarantee this condition, all inactive a-nodes must be deleted after the network settles on the attachments for each phrase. Otherwise, losing a-nodes could become activated later in the parse, when the network is no longer in a configuration in which they compete with their incompatible alternatives. Since losing a-nodes are deleted, CAPERS maintains only a single valid parse state at any time.

The use of CBSA, and the adoption of a stack mechanism to support this, strongly restrict the attachments that can be considered by the parser. The only a-nodes that can compete simultaneously are those in the set of attachments between the current phrase and the tree on top of the stack. The competitive

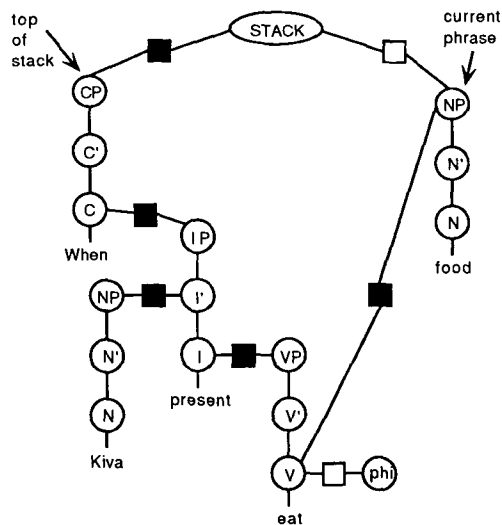


Figure 10: The NP *food* has a single valid attachment to the parse tree.

has revised its earlier attachment hypothesis for the NP and incorporated the new IP phrase into the parse tree.

Sentence (2), an example of Late Closure effects, is initially processed in a similar fashion:

(2) When Kiva eats food gets thrown.

After attaching *food*, the network has the configuration shown in Figure 10. As in sentence (1), the post-verbal NP makes the best attachment available to it, as the complement of the verb. This behavior is again consistent with the initial preferences of the human sentence processor (Frazier & Rayner 82). Since the initial attachment in these cases of Late Closure is determined in exactly the same manner as the Minimal Attachment cases illustrated by sentence (1), these two classic preferences receive a uniform account in the CAPERS model.

Additional processing of the input distinguishes the sentence types. At *gets*, a sentential phrase is projected, and the network settles on the attachments shown in Figure 11. As in Figure 8, the revision necessary for a valid parse involves the current phrase and the right edge of the tree. However, in this case, the misattached NP cannot break its attachment to the verb and reattach as the specifier of the IP. The difference from the prior example is that here the V node has no other a-node to redirect its output to, and so it continues to activate the NP attachment. The attachment of the NP to the I' is not strong enough by itself to draw activation away from the attachment of the NP to the V. The current I' thus activates the default phi-node attachment, leading to a clause with

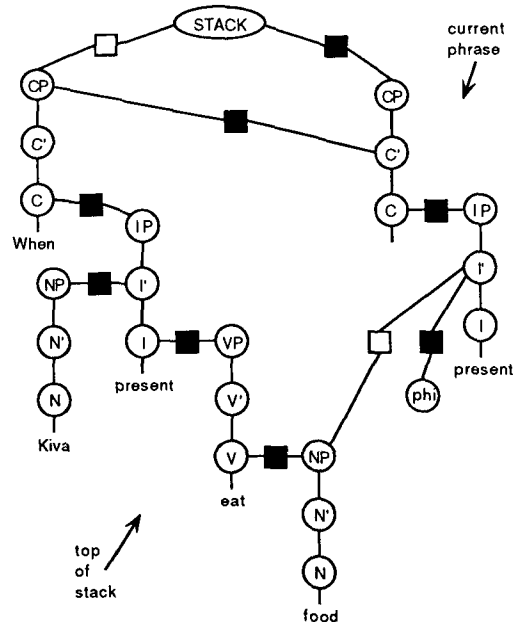


Figure 11: The attachment of the NP *food* to the I' is not strong enough to break the attachment of the NP to the V.

an empty (and unbound) subject. Since the network settles on an irrecoverably ungrammatical analysis, CAPERS correctly predicts a garden path.

The next two examples, adapted from (Pritchett 88), involve double object verbs; both types of sentences clearly garden path the human sentence processor. In each case, the second post-verbal NP is the focus of attention. In sentence (3), this NP is the subject of a relative clause modifying the first NP, but the parser misinterprets it as the verb's second complement:

(3) Jamie gave the child the dog bit a bandaid.

The initial connections of the NP *the dog* to the network are shown in Figure 12. The NP can either push itself onto the stack, or replace the null attachment of the verb to the phi-node. Since both stack attachments and phi-node attachments are relatively weak, the NP attachment to the V wins the a-node competition, and the network settles on the tree in Figure 13. In accordance with human preferences, the NP is attached as the second object of the verb. When *bit* is processed, the network settles on the configuration in Figure 14. As in the earlier examples, the misattached NP needs to attach as the subject of the current clause; however, this would leave the V node with only one a-node to activate instead of its required two attachments. CAPERS again settles on an ungrammatical analysis in which the current clause has an

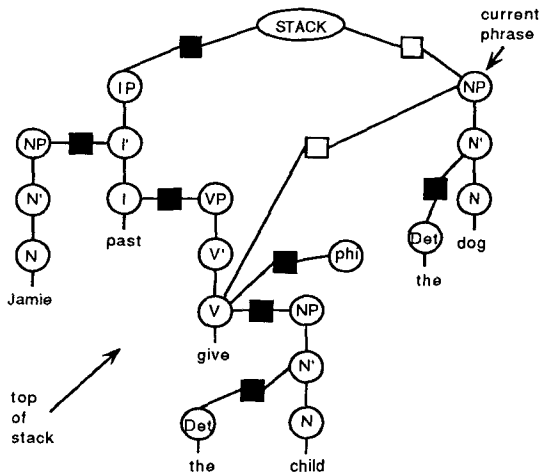


Figure 12: The initial connections of the NP *the dog* to the network.

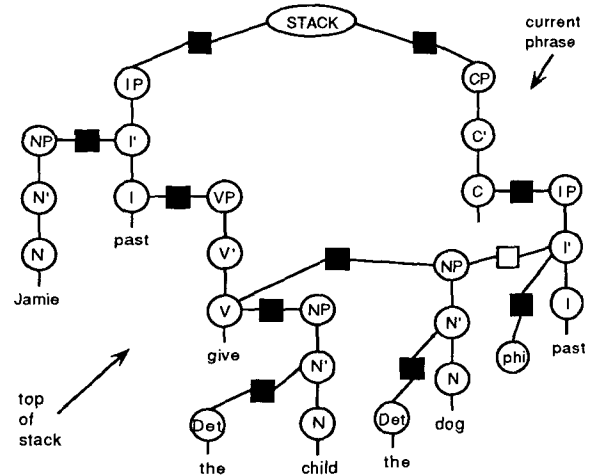


Figure 14: If the NP *the dog* activates the attachment to the I', the V node would be left with only one active attachment.

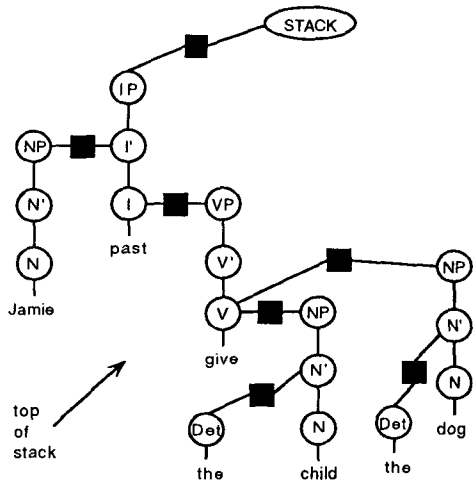


Figure 13: The NP *the dog* attaches as the verb's second complement.

empty (unbound) subject, consistent with the garden path effect of this sentence.

The second example with a double object verb involves the opposite problem. In sentence (4), the second post-verbal NP is mistakenly interpreted as part of the first object; in a complete parse, it is part of the second object:

- (4) I convinced her children are noisy.

Initially, the parser attaches *her* as the NP object of *convinced*. The structure of the network after attachment of *children* is shown in Figure 15. The NP *children* cannot replace the phi-node attachment to the verb, since the second object of *convince* must be

sentential. In order to maximally satisfy the attachment preferences, *her* is reanalyzed as the specifier of *children*, with *her children* replacing *her* as the first object of *convinced*. This reanalysis is structurally the same as that required in Figure 8; the relevant a-nodes have been numbered the same in each figure to highlight the similarity. Problems arise when the network attaches the next input word, *are*; see Figure 16. Once again, the misattached NP needs to attach as the specifier of the following sentential phrase, but a V node would be left with only one active a-node when it requires two. A garden path once more results from the network settling on an ungrammatical analysis.

This example highlights another aspect of the competitive mechanism of CAPERS in driving the attachment behavior of the parser: the only way a previous attachment can be broken is if it participates in a competition with an attachment to the current phrase. A correct parse requires *her* to break its attachment to *children* and re-attach directly to the verb. Because the a-node attaching *her* to *children* has no competitor, there is no mechanism for changing the problematic attachment.

5 Summary

In each of the examples of Section 4, the initial attachment of a phrase was incompatible with the remainder of the sentence. CAPERS can recover from an attachment error of this type exactly when the misattached phrase can reattach to the current phrase, with the current phrase "replacing" the misattached

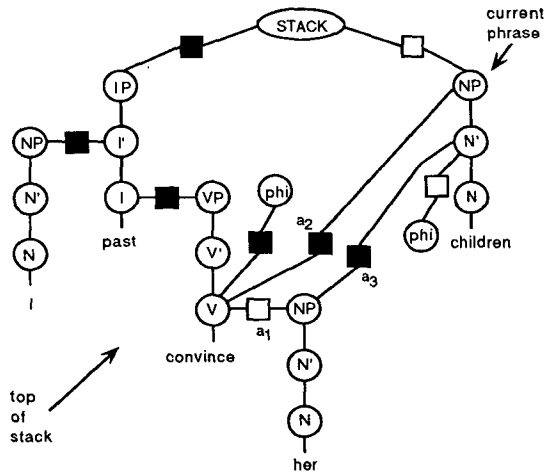


Figure 15: Attaching the NP *children* requires reanalysis of the NP *her*.

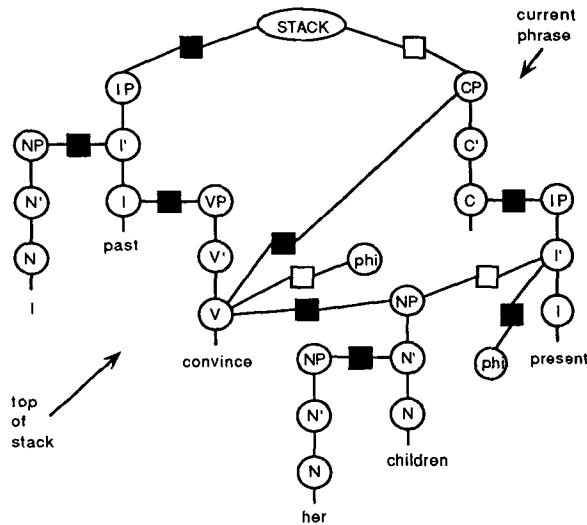


Figure 16: If the NP headed by *children* activates the attachment to the I', the V node would be left without an NP complement.

phrase in its original attachment site. If the p-node to which the misattached phrase was originally attached does not have an alternative a-node to activate, reanalysis cannot take place and a garden path results. The allowable attachment configurations are a direct consequence of the restrictions imposed by the competitive mechanism of CAPERS. The resulting initial attachment preferences, and the parser's ability or inability to revise the incorrect structure, account for the preferred readings of these temporarily ambiguous sentences, as well as the garden path results.

References

- Abney, S. (1986). "Functional elements and licensing." GLOW Conference, Gerona, Spain.
- Abney, S. (1989). "A computational model of human parsing." *Journal of Psycholinguistic Research* 18:1, 129-144.
- Chomsky, N. (1981). *Lectures on Government and Binding: The Pisa Lectures*. Dordrecht: Foris Publications.
- Chomsky, N. (1986). *Barriers*. Cambridge: MIT Press.
- Cottrell, G. W. (1989). *A Connectionist Approach to Word Sense Disambiguation*. Los Altos, CA: Morgan Kaufmann.
- Fant, M. (1985). "Context-free parsing in connectionist networks." Technical Report TR174, University of Rochester.
- Frazier, L. (1978). *On Comprehending Sentences: Syntactic Parsing Strategies*. Doctoral dissertation, University of Connecticut. Bloomington, IN: Indiana University Linguistics Club.
- Frazier, L., and K. Rayner (1982). "Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences." *Cognitive Psychology* 14, 178-210.
- Gibson, E. (1991). "A Computational Theory of Human Linguistic Processing: Memory Limitations and Processing Breakdown." Doctoral dissertation, Carnegie-Mellon University.
- Gorrell, P. (1987). "Studies of Human Syntactic Processing: Ranked-Parallel versus Serial Models." Unpublished doctoral dissertation, University of Connecticut, Storrs, CT.
- Inoue, A. and J. Fodor (1992). "Information-paced parsing of Japanese." Presented at the Fifth Annual CUNY Conference on Human Sentence Processing, New York.
- McRoy, S. and G. Hirst (1990). "Race-Based Parsing and Syntactic Disambiguation." *Cognitive Science* 14, 313-353.
- Pritchett, B. (1988). "Garden Path Phenomena and the Grammatical Basis of Language Processing." *Language* 64:3, 539-576.
- Rizzi, L. (1990). *Relativized Minimality*. Cambridge: MIT Press.
- Reggia, J. (1987). "Properties of a Competition-Based Activation Mechanism in Neuromimetic Network Models." *Proceedings of the First International Conference on Neural Networks*, San Diego, II-131-II-138.
- Reggia, J., P. Marsland, and R. Berndt (1988). "Competitive Dynamics in a Dual-Route Connectionist Model of Print-to-Sound Transformation." *Complex Systems*.
- Selman, G., and G. Hirst (1985). "A Rule-Based Connectionist Parsing Scheme." *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, 212-219.
- Shieber, S. (1983). "Sentence Disambiguation by a Shift-Reduce Parsing Technique." *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, 113-118.
- Stevenson, S. (1993a). "Establishing Long-Distance Dependencies in a Hybrid Network Model of Human Parsing." *Proceedings of the 15th Annual Conference of the Cognitive Science Society*.
- Stevenson, S. (1993b). "A Constrained Active Attachment Model for Resolving Syntactic Ambiguities in Natural Language Parsing." Doctoral dissertation, Computer Science Department, University of Maryland, College Park.
- Stevenson, S. (1990). "A Parallel Constraint Satisfaction and Spreading Activation Model for Resolving Syntactic Ambiguity." *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, 396-403.