Title

A Parallel Model For Adult Sentence Processing

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 10(0)

Authors

Clacrk, Robin
Gibson, Edward

Publication Date

1988

Peer reviewed

A PARALLEL MODEL FOR ADULT SENTENCE PROCESSING[1]

ROBIN CLARK AND EDWARD GIBSON

CARNEGIE MELLON UNIVERSITY

1. INTRODUCTION

Serial deterministic models of human sentence processing (e.g. Frazier, 1978; Frazier & Fodor, 1978; Marcus, 1980; Berwick, 1985) have had a great intuitive appeal, especially when contrasted with unlimited parallel processing. A serial deterministic parser, for example, provides a straightforward account of why people often fail to perceive ambiguity and why they report a strongly favoured interpretation for ambiguous sentences. Serial deterministic processing, furthermore, gives a very plausible account of how people are misled into a garden-path parse for certain strings:

(1) The horse raced past the barn fell.

By virtue of returning all possible grammatical parses for an input sentence, an unlimited parallel parser, by itself, gives us no reason to prefer one parse over another, and, prima facie, such a model fails to account for the preferred readings of ambiguous input. Because of its computational power, a parallel parser cannot be misled, since the non-preferred analysis of garden-path input can be carried along by the parallelism. Hence, a fully parallel parser provides no obvious account of garden-path phenomena.

Parallel models of processing, on the other hand, can provide a simple account of relative processing load.[2] When the parser encounters ambiguous input, the number of hypotheses entertained by the parser will increase and more computational resources (memory and time measured in number of operations) will be used. Since a deterministic model is committed to a single hypothesis at all times (Marcus, 1980; Berwick, 1985), the computational resources used by the parser should remain virtually constant. A deterministic model, therefore, has no obvious way of modeling variations of processing load with respect to ambiguous input (Gorrell, 1986).

Constrained parallel processing may provide an account of garden-path effects and preferred readings for ambiguous input, on the one hand, and variations in the relative complexity of processing, on the other. As long as a parallel model is limited so that it obtains garden-path and other classic psycholinguistic effects generally associated with serial models, it is a valid model. Until recently, however, serial models have been generally preferred to parallel ones, since it was not obvious how to constrain the power of a fully parallel model.

It has been observed in a number of studies that word recognition is parallel (Swinney, 1979; Tanenhaus, Leiman & Seidenberg, 1979; Seidenberg, Waters, Sanders & Langer, 1984). Since syntactic processing makes use of the results obtained from word recognition, a deterministic syntactic processor may select only one lexical item from this list. It is unclear how to constrain the lexical decision task imposed on the parser without recourse to devices such as look-ahead or multiple passes over the input string (cf., the "attention shifts" of Marcus, 1980). Evidence indicates that lexical selection is not sensitive to pragmatic knowledge (Swinney, 1979). Following Fodor (1983),

[1]We would like to thank Rick Kazman for his comments on an earlier draft of this paper.

[2]On variations in relative complexity of processing, see Fodor, Garrett & Bever, 1968; Holmes & Forster, 1972; Hakes, 1972; Rayner, Carlson & Frazier, 1983; Frazier & Rayner, 1987; Shapiro, Zurif & Grimshaw, 1987, among others.

we assume that syntactic processing is "informationally encapsulated"; that is, the syntactic processor does not make appeal to extra-linguistic pragmatic knowledge. Furthermore, we assume that syntactic processing is automatic in that it is fast and not subject to conscious manipulation. A parallel syntactic processor offers great uniformity between lexical recognition and sentence processing. Parallel syntactic processing may also illuminate the difficult task of selecting the syntactically appropriate reading of ambiguous lexical items without appeal to problematic constructs like look-ahead and multiple passes by the parser. Recently, psycholinguists have reconsidered parallelism in the context of sentence processing (Kurtzman, 1985; Gorrell, 1986; Carlson & Tanenhaus, 1986), although few explicit parallel models have been formulated. We will describe a parallel algorithm for sentence processing which is compatible with recent research in psycholinguistics in that it produces well-known effects; furthermore, since the model is explicitly formalized, its predictions are readily subject to empirical disconfirmation.

In this paper we present Constrained Parallel Parser (CPP), a parser based on the principles of Government-Binding Theory (Chomsky, 1981; Chomsky, 1986); crucially, CPP has no separate grammar rule module containing language-particular rules. Unlike many psycholinguistic models, CPP has been implemented; it exists as a CommonLISP program. In the CPP model, distinct tree structures correspond to different argument structures of a given lexical entry. When a word is input, these representations are built and placed in the *buffer*, a one cell data structure that holds a parallel list of these representations. Hence, the complexity of the parse will be contingent on the number of argument structures associated with a lexical item (cf., Shapiro, Zurif & Grimshaw, 1987). CPP contains a second data structure, the *stack*, that is of the same structure as the buffer, but may be more than one cell deep. The parser builds trees in parallel based on possible attachments made between the buffer and the top of the stack. In addition, the attachments are limited by constraints on the algorithm and constraints on the resulting representations. Both types of constraints can be shown to cause garden-path effects (see Clark, 1987; Gibson, 1987).

2. LEXICAL ENTRIES FOR CPP

A lexical entry accessed by CPP consists in part of a *theta-grid*. A theta-grid is an unordered list of *theta structures*. Each theta structure consists of a thematic role and associated subcategorization information. One theta structure in a theta-grid may be marked as *indirect* to indicate that it must be assigned to the subject of the phrase. For example, the word *shout* might have the following theta-grid:[3]

```
((Subcat = NOUN, Thematic-Role = AGENT, INDIRECT)
 (Subcat = PREP, Thematic-Role = GOAL)
 (Subcat = COMP, Thematic-Role = PROPOSITION))
```

When the word *shout* [4] is encountered in an input phrase, the thematic role *agent* will be assigned to its subject, as long as this subject is a noun phrase. The direct thematic roles *goal* and *proposition* will be assigned to prepositional and complementizer phrases respectively, as long as each is present. Since the order of theta structures in a theta-grid is not relevant to its use in parsing, the above theta-grid for *shout* will be sufficient to parse both sentences (2) and (3).

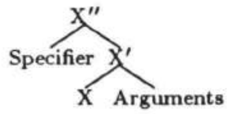
[3] In a more complete theory, a syntactic category would be determined from the thematic role.

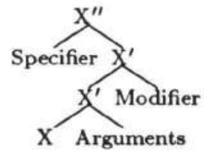
[4] In the current system, morphological variation is only covered by explicit listing of lexical entries. A morphological analysis component is proposed to extend the current system.

(2) `The man shouts to the woman that Ernie sees the rock.`
(3) `The man shouts that Ernie sees the rock to the woman.`

3. THE CPP ALGORITHM

The CPP model assumes $\overline{\text{X}}$ Theory as present in Chomsky (1986). Each tree structure must have a head and each head must have a maximal projection. These principles interact with other principles built into the parser, (*e.g.*, the θ–Criterion, the Extended Projection Principle, Case Theory) to determine the positions of arguments, specifiers and modifiers with respect to the head of a given structure. As a result, a specifier may only appear as a sister to the one-bar projection below a maximal projection, and the head must appear below the one-bar projection along with its arguments. For example, the structure for categories in English is shown on the left below, with a modifier attachment on the left below.





The Constrained Parallel Parser is a head-driven parser that builds structure by making attachments between the buffer and the top of the stack. For each lexical entry, a maximal projection of that entry is placed in the buffer. The *buffer* is a one cell data structure that contains a set of tree structures, each of which represents the same segment of the input string. The data structure *stack* consists of a stack of buffer cells.

Since CPP has no separate grammar rule module containing language-specific rules, an attachment between a structure in the buffer and a structure on the stack is possible based the configuration of the given parser. This parser contains constants that are independent of the language being parsed along with parameters that depend on the language being parsed. For example, the fact that determiners, if they exist as an independent category in a given language, are attached in the specifier position of NP follows from Universal Grammar. Whether this attachment takes place from stack to buffer or from buffer to stack varies according to the type of attachment and language being considered, however. In English, specifier attachment takes place from stack to buffer, indicating that, in English, specifiers occur before the head. As a result, a parameter would be set in a parser for English that indicates specifier attachment occurs from stack to buffer. Arguments, on the other hand, are attached from buffer to stack in English, since English is head-first with respect to arguments. As with the case of Specifiers, this order is the result of setting a parameter which dictates the direction of the attachment of complements with respect to the head.

The formal CPP algorithm is given below, with parameters for attachment set to parse English.

1. (Initializations) Set the STACK to NIL. Set the BUFFER to NIL.
2. (Ending Condition) If the input string is finished and the BUFFER is empty then return the contents of the STACK and stop.
3. If the BUFFER is empty then create a maximal projection for each lexical entry corresponding to the next word in the input string, and put this list of maximal projections into the BUFFER.

4. Make all possible attachments between the STACK and the BUFFER, subject to the attachment constraints. Put the attached structures in the BUFFER. If no attachments are possible, then put the contents of the BUFFER on top of the STACK.
5. Go to 2.

Possible Attachments (parameterized for English):

- Argument Attachment: (BUFFER to STACK) If a structure B in the BUFFER is compatible with the lexical requirements of a structure A, on top of the STACK, then attach B to A as an argument.
- Specifier Attachment: (STACK to BUFFER) If a structure A, on top of the STACK, is compatible as a specifier of a structure B, in the BUFFER, then attach A to B as a specifier.
- Pre-Head Adjunct Attachment: (STACK to BUFFER) If a structure A, on top of the STACK, is compatible as a modifier of a structure B, in the BUFFER, then attach A to B as a modifier.
- Post-Head Adjunct Attachment: (BUFFER to STACK) If a structure B in the BUFFER is compatible as a modifier of a structure A, on top of the STACK, then attach B to A as an modifier.

Attachment Constraints:

- **Exclusive Attachment Constraint**: If an attachment is possible between two structures (one on the stack, one in the buffer), then it is made. All nodes in parallel that do not take part in attachment, either on the stack or in the buffer, are pruned.
- **Case Filter and θ-Criterion**[5]: After attachments between stack and buffer are completed, if a structure A directly receives a necessary property (*e.g.*, a thematic role, abstract Case for a lexical noun phrase), then prune all representations in which structure A appears but does not directly receive that property. For example, if a certain lexical noun phrase NP_1 receives Case in one representation, then all representations are pruned in which NP_1 does not receive Case.
- **Lexical Requirement Constraint**: If an attachment is possible that satisfies the lexical requirements of some head, then make that attachment and all others that also satisfy lexical requirements. If no such attachment is possible, then make any other possible attachments satisfying other constraints.

The Exclusive Attachment Constraint (EAC) guarantees that each of the parallel structures in a buffer cell represents the same segment of the input string. This uniformity permits the use of simple data structures. The EAC is extended into the Principles of Government-Binding theory to give the CPP versions of the Case Filter and θ-Criterion. The Lexical Requirement Constraint (LRC) is a parsing extension of the Projection Principle[6]. As a result of these constraints, garden-path and other psycholinguistic effects occur.

The CPP implementation is on a serial machine, so the parallelism must be mimicked. The parser still runs quite fast, averaging about one-tenth of a second per word on a Hewlett Packard 9000/350 with 8 megabytes of RAM. This speed can be partially attributed to the empirically observed fact that most parallelism dies very quickly using the CPP algorithm.

[5]The Case Filter states that a lexical noun phrase must receive abstract Case. The θ-Criterion states that all arguments must receive exactly one thematic role and that all thematic roles must be assigned.

[6]The Projection Principle states that all lexical requirements must be satisfied.

4. GARDEN-PATH EFFECTS AND CPP

To illustrate the algorithm in action, consider the garden-path sentence, sentence (4):

(4) The man walked to the station ate the cake.

This is a garden-path sentence because *walked to the station* is misanalyzed as matrix level verb phrase: to obtain a grammatical sentence, *walked to the station* must be analyzed as a reduced relative clause modifying the noun phrase *the man*.

The parse begins with the placement of a maximal projection for the determiner *the* in the buffer. Since there is nothing on the stack, no attachments can be made, and the determiner phrase simply moves to the stack. The second word, *man*, is then read from the input string. Since *man* has both noun and verb entries in the lexicon, a maximal projection for each reading enters the buffer, as shown below:

BUFFER: ($[_{N''}$ $[_{N'}$ $[_{N}$ *man*]]]
$[_{V''}$ $[_{V'}$ $[_{V}$ *man*]]])
STACK: (($[_{Det''}$ $[_{Det'}$ $[_{Det}$ *the*]]]))

We note that, at this point, the relative processing load has increased due to the ambiguity of *man*. Now that both the stack and buffer are non-empty, attachments may be tried. Argument attachment fails, since the structure on top of the stack, the determiner phrase representing *the*, has no lexical requirements. Since a determiner cannot modify or be modified by either a noun or verb phrase, both pre- and post-head modifier attachments fail. Specifier attachment fails between the determiner and the verb, but succeeds between the determiner and the noun, since a determiner is a possible specifier for a noun phrase.[7] The noun phrase resulting from attachment is then placed in the buffer. Since the verb phrase reading of *man* did not take place in the attachment, it is pruned from the parse by the Exclusive Attachment Constraint. At this point, the complexity of the parse decreases, since one representation has been eliminated. The contents of the buffer are then moved onto the stack and maximal projections for the next input word, *walked*, are placed in the buffer. These projections consist of a verb phrase representing the passive participle *walked*, as well as an Infl phrase representing the tensed verb *walked*.[8]

BUFFER: ($[_{Infl''}$ *walked*]
$[_{V''}$ *walked*]
STACK: (($[_{N''}$ *the man*]))

Argument attachment fails, since the noun phrase *the man* has no lexical requirements. Specifier attachment succeeds between the Infl phrase and the noun phrase, as well as between the verb phrase and the noun phrase. Modifier attachment also succeeds between the noun phrase in the buffer and the verb phrase on the stack. Since nominative Case is assigned to the NP specifier of the Infl phrase attachment, the Case Filter is activated. No Case is assigned to the NP *the man* in the small clause verb phrase reading or in the noun phrase with modifier reading, so these two

[7] This is presumably a theorem of Universal Grammar. That is, a determiner may be attached as the Specifier of a noun phrase. Thus, provided that a language has determiners, they will be attached as [Spec, N]. We assume, furthermore, that UG allows only this role for determiners; they cannot be modifiers of VP for example.

[8] The category *Infl* contains inflection information. An Infl phrase is automatically built for any tensed verb, since tense resides in Infl in the GB framework.

representations are pruned. As a result, the reduced relative clause reading of the phrase *walked to the station* is ignored, which eventually leads to the garden-path effect. Maximal projections for the word *to* now enter the buffer: a prepositional phrase and Infl phrase. Both attach as modifiers to the matrix verb phrase. The determiner *the* then enters the buffer. No attachments are possible at this point, so the contents of the buffer are pushed onto the stack. A maximal projection for the noun *station* is then placed in the buffer and the determiner on top of the stack attaches to it. The state of the parse at this point is as follows:

BUFFER: ($[_{Infl''}$ $[_{N''}$ *the man*] $[_{Infl'}$ $[_{V''}$ $[_{V'}$ $[_{V'}$ *walked* $[_{P''}$ *to*]]]]]]
$[_{Infl''}$ $[_{N''}$ *the man*] $[_{Infl'}$ $[_{V''}$ $[_{V'}$ $[_{V'}$ *walked* $[_{Comp''}$ $[_{Infl''}$ *to*]]]]]]]
STACK: (($[_{N''}$ *the station*]))

Only argument attachment succeeds for the prepositional reading of *to*, while no attachment is possible for the Infl reading of *to*. The Exclusive Attachment Constraint therefore prunes the structure containing the embedded Infl reading.

The tensed Infl phrase *ate* now enters the buffer. No attachments are possible between the two Infl phrases and the Extended Projection Principle[9] is violated. The garden-path effect follows naturally from general constraints on the parallelism displayed by the parser. These same constraints are partially responsible for the observed speed of the parser.

5. CONCLUSIONS

We have described a parallel parsing model that, like serial deterministic models, obtains garden-path effects. The design of the parser follows from current work in syntactic theory. The representations posited by the parser must obey certain constraints (the Case Filter, θ-Criterion, etc.). Furthermore, in keeping with the spirit of recent work in Government-Binding theory (Stowell, 1981; Chomsky, 1985), the parser makes no use of language-particular grammar rules. The absence of language-particular rules, in conjunction with constraints on parallelism are responsible for much of the speed of the parser. Given the parser's ability to replicate phenomena like garden-path effects, we feel that research along these lines can do much to illuminate the relationship between knowledge and its use. Finally, we note that the Constrained Parallel Parser is a genuinely parallel parser. Unlimited parallel parsers cannot obtain garden-path effects, because of their inability to err. The CPP model, since it is severely constrained, does not suffer this defect. Hence, psycholinguistic theory, while correct in abandoning unconstrained parallelism, stands to profit from the study of constrained parallel algorithms. Finally, the CPP algorithm provides a non-trivial alternative to standard methods of parsing that make use of charts, networks or case-frames; the algorithm has the potential of yielding an interesting formal basis for the empirical study of adult sentence processing.

REFERENCES

Berwick, R. (1985), *The Acquisition of Syntactic Knowledge*, MIT Press, Cambridge, MA.

Carlson, G.N., & Tanenhaus, M.K. (1987), *Thematic Roles and Language Comprehension*, University of Rochester manuscript.

[9]The Extended Projection Principle (Chomsky, 1981) states that all lexical requirements must be satisfied and that all Infl phrases must have subjects.

Chomsky, N. (1981), *Lectures on Government and Binding*, Foris, Dordrecht, The Netherlands.

Chomsky, N. (1985), *Knowledge of Language: Its Nature, Origin and Use*, Praeger Publishers, New York, NY.

Chomsky, N. (1986), *Barriers*, Linguistic Inquiry Monograph 13, MIT Press, Cambridge, MA.

Clark, R. (1987), *Rules and Parsing*, Talk presented at MIT.

Fodor, J.A. (1983), *Modularity of Mind*, MIT Press, Cambridge, MA.

Fodor, J.A., Garrett, M.F., & Bever, T.G. (1968), *Some Syntactic Determinants of Sentential Complexity*, Perception and Psychophysics 2, pp. 289-96.

Frazier, L., & Fodor, J.D. (1978), *The Sausage Machine: A New Two-stage Parsing Model*, Cognition 6, pp. 291–325.

Frazier, L. (1978), *On Comprehending Sentences: Syntactic Parsing Strategies*, University of Massachusetts Ph.D. dissertation. Frazier, L. (1985), *Syntactic Complexity*, from Dowty, Karttunen & Zwicky (eds.), *Natural Language Parsing*, Cambridge University Press.

Frazier, L. & Rayner, K. (1987), *Resolution of Syntactic Category Ambiguities: Eye Movements in Parsing Lexically Ambiguous Sentences*, Journal of Memory and Language 26.

Gibson, E.A.F. (1987), *Garden-Path Effects in a Parser with Parallel Architecture*, Eastern States Conference on Linguistics.

Gorrell, P.G. (1987), *Studies of Human Syntactic Processing: Ranked-Parallel versus Serial Models*, University of Connecticut Ph.D. dissertation.

Hakes, D.T. (1972), *Effects of Reducing Complement Constructions on Sentence Comprehension*, Journal of Verbal Learning and Verbal Behavior 11, pp. 278-86.

Holmes, V.M. & Forster, K.I. (1972), *Perceptual Complexity and Understanding Sentence Structure*, Journal of Verbal Learning and Verbal Behavior 11, pp. 148-56.

Jackendoff, R. (1977), *X-bar Syntax: A Study of Phrase Structure*, Linguistic Inquiry Monograph 2, MIT Press, Cambridge, MA.

Kurtzman, H. (1985), *Studies in Syntactic Ambiguity Resolution* MIT Ph.D. dissertation.

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

Rayner, K., Carlson, M., & Frazier, L. (1983), *The Interaction of Syntax and Semantics during Sentence Processing: Eye Movements in the Analysis of Semantically Biased Sentences*, Journal of Verbal Learning and Verbal Behavior, 22, pp 358-374.

Seidenberg, M.S., Waters, G.S., Sanders, M., & Langer, P. (1984), *Pre- and Post-Lexical Loci of Contextual Effects on Word Recognition*, Memory and Cognition, 12, 315-328.

Shapiro, L., Zurif, E. & Grimshaw, J. (1987), *Sentence Processing and the Mental Representation of Verbs*, Cognition 29.

Stowell, T. (1981), Origins of Phrase Structure, MIT Ph.D. dissertation.

Swinney, D.A. (1979), *Lexical Access during Sentence Comprehension: (Re)consideration of Context Effects*, Journal of Verbal Learning and Verbal Behavior, 18, pp 549-569.

Tanenhaus, M.K., Leiman, J.M., & Seidenberg, M.S. (1979), *Evidence for Multiple Stages in the Processing of Ambiguous Words in Syntactic Contexts* Journal of Verbal Learning and Verbal Behavior, 18, pp 427-440.