On Parsing Strategies and Closure[1]

Kenneth Church
MIT
Cambridge, MA 02139

This paper proposes a welcome hypothesis: a computationally simple device[2] is sufficient for processing natural language. Traditionally it has been argued that processing natural language syntax requires very powerful machinery. Many engineers have come to this rather grim conclusion; almost all working parsers are actually Turing Machines (TM). For example, Woods believed that a parser should have TM complexity and specifically designed his Augmented Transition Networks (ATNs) to be Turing Equivalent.

(1) "It is well known (cf. [Chomsky64]) that the strict context-free grammar model is not an adequate mechanism for characterizing the subtleties of natural languages." [Woods70]

If the problem is really as hard as it appears, then the only solution is to grin and bear it. Our own position is that parsing acceptable sentences is simpler because there are constraints on human performance that drastically reduce the computational complexity. Although Woods correctly observes that competence models are very complex, this observation may not apply directly to a performance problem such as parsing.[3]

The claim is that performance limitations actually reduce parsing complexity. This suggests two interesting questions: (a) *How is the performance model constrained so as to reduce its complexity*, and (b) *How can the constrained performance model naturally approximate competence idealizations?*

1. The FS Hypothesis

We assume a severe processing limitation on available short term memory (STM), as commonly suggested in the psycholinguistic literature ([Frazier79], [Frazier and Fodor79], [Cowper76], [Kimball73, 75]). Technically a machine with limited memory is a finite state machine (FSM) which has very good complexity bounds compared to a TM.

How does this assumption interact with competence? It is plausible for there to be a rule of competence (call it $C_{complex}$) which cannot be processed with limited memory. What does this say about the psychological reality of $C_{complex}$? What does this imply about the FS hypothesis?

When discussing certain performance issues (e.g. center-embedding),[4] it will be most useful to view the processor as a FSM; on the other hand, competence phenomena (e.g. subjacency) suggest a more abstract point of view. It will be assumed that there is ultimately a single processing machine with its multiple characterizations (the ideal and the real components). The processor does not literally apply ideal rules of competence for lack of ideal TM resources, but rather, it resorts to more realistic approximations. Exactly where the idealizations call for inordinate resources, we should expect to find empirical discrepancies between competence and performance.

A FS processor is unable to parse complex sentences even though they may be grammatical. We claim these complex sentences are unacceptable. Which constructions are *in principle* beyond the capabilities of a finite state machine? Chomsky and Bar-Hillel independently showed that (arbitrarily deep) center-embedded structures require unbounded memory [Chomsky59a, b] [Bar-Hillel61] [Langendoen75]. As predicted, arbitrarily center-embedded sentences are unacceptable, even at relatively shallow depths.

(2) #[The man [who the boy [who the students recognized] pointed out] is a friend of mine.]

(3) #[The rat [the cat [the dog chased] bit] ate the cheese.]

A memory limitation provides a very attractive account of the center-embedding phenomena (in the limit).[5]

(4) "This fact [that deeply center-embedded sentences are unacceptable], and this alone, follows from the assumption of finiteness of memory (which no one, surely, has ever questioned)." [Chomsky61, pp. 127]

What other phenomena follow from a memory limitation? Center-embedding is the most striking example, but it is *not* unique. There have been many refutations of FS competence

1. I would like to thank Peter Szolovits, Mitch Marcus, Bill Martin, Bob Berwick, Joan Bresnan, Jon Allen, Ramesh Patil, Bill Swartout, Jay Keyser, Ken Wexler, Howard Lasnik, Dave McDonald, Per-Kristian Halvorsen, and countless others for many useful comments.

2. Throughout this work, the complexity notion will be used in its computational sense as a measure of time and space resources required by an optimal processor. The term will not be used in the linguistic sense (the size of the grammar itself). In general, one can trade one off for the other, which leads to considerable confusion. The size of a program (linguistic complexity) is typically inversely related to the power of the interpreter (computational complexity).

3. A hash mark (#) is used to indicate that a sentence is unacceptable; an asterisk (*) is used in the traditional fashion to denote ungrammaticality. Grammaticality is associated with competence (post-theoretic), whereas acceptability is a matter of performance (empirical).

4. A center-embedded sentence contains an embedded clause surrounded by lexical material from the higher clause: $[_S\ x\ [_S\ \_]\ y]$, where both x and y contain lexical material.

5. A complexity argument of this sort does not distinguish between a depth of three or a depth of four. It would require considerable psychological experimentation to discover the precise limitations.

models; each one illustrates the point: *computationally complex structures are unacceptable.* Lasnik's noncoreference rule [Lasnik76] is another source of evidence. The rule observes that two noun phrases in a particular structural configuration are noncoreferential.

(5) The Noncoreference Rule: Given two noun phrases NP_1, NP_2 in a sentence, if NP_1 precedes and commands NP_2 and NP_2 is not a pronoun, then NP_1 and NP_2 are noncoreferential.

It appears to be impossible to apply Lasnik's rule with only finite memory. The rule becomes harder and harder to enforce as more and more names are mentioned. As the memory requirements grow, the performance model is less and less likely to establish the noncoreferential link. In (6), the co-indexed noun phrases cannot be coreferential. As the depth increases, the noncoreferential judgments become less and less sharp, even though (6)-(8) are all equally ungrammatical.

(6) *#Did you hear that John$_i$ told the teacher John$_i$ threw the first punch.
(7) *??Did you hear that John$_i$ told the teacher that Bill said John$_i$ threw the first punch.
(8) *?Did you hear that John$_i$ told the teacher that Bill said that Sam thought John$_i$ threw the first punch.

Ideal rules of competence do not (and should not) specify real processing limitations (e.g. limited memory); these are matters of performance. (6)-(8) do not refute Lasnik's rule in any way; they merely point out that its performance realization has some important empirical differences from Lasnik's idealization.

Notice that movement phenomena can cross unbounded distances without degrading acceptability. Compare this with the center-embedding examples previously discussed. We claim that center-embedding demands unbounded resources whereas movement has a bounded cost (in the worst case).[6] It is possible for a machine to process unbounded movement with very limited resources.[7] This shows that movement phenomena (unlike center-embedding) can be implemented in a performance model *without approximation.*

(9) There seems likely to seem likely ... to be a problem.
(10) What did Bob say that Bill said that ... John liked?

It is a positive result when performance and competence happen to converge, as in the movement case. Convergence enables performance to apply competence rules without approximation. However, there is no logical necessity that performance and competence will ultimately converge in every area. The FS hypothesis, if correct, would necessitate compromising many competence idealizations.

2. The Proposed Model: YAP

Most psycholinguists believe there is a natural mapping from the complex competence model onto the finite performance world. This hypothesis is intuitively attractive, even though there is no logical reason that it need be the case.[8] Unfortunately, the psycholinguistic literature does not precisely describe the mapping. We have implemented a parser (YAP) which behaves like a complex competence model on acceptable[9] cases, but fails to parse more difficult unacceptable sentences. This performance model looks very similar to the more complex competence machine on acceptable sentences even though it "happens" to run in severely limited memory. Since it is a minimal augmentation of existing psychological and linguistic work, it will hopefully preserve their accomplishments, and in addition, achieve computational advantages.

The basic design of YAP is similar to Marcus' Parsifal [Marcus79], with the additional limitation on memory. His parser, like most stack machine parsers, will occasionally fill the stack with structures it no longer needs, consuming unbounded memory. To achieve the finite memory limitation, it must be guaranteed that this never happens on acceptable structures. That is, there must be a procedure (like a garbage collector) for cleaning out the stack so that acceptable sentences can be parsed without causing a stack overflow. Everything on the stack should be there for a reason; in Marcus' machine it is possible to have something on the stack which cannot be referenced again. Equipped with its garbage collector, YAP runs on a bounded stack even though it is approximating a much more complicated machine (e.g. a PDA).[10] The claim is that YAP can parse acceptable sentences with limited memory, although there may be certain unacceptable sentences that will cause YAP to overflow its stack.

3. Marcus' Determinism Hypothesis

The memory constraint becomes particularly interesting when it is combined with a control constraint such as Marcus' Determinism Hypothesis [Marcus79]. The Determinism Hypothesis claims that once the processor is committed to a particular path, it is extremely difficult to select an alternative. For example, most readers will misinterpret the underlined portions of (11)-(13) and then have considerable difficulty continuing. For this reason, these unacceptable sentences are often called Garden Paths (GP). The memory limitation alone fails to predict the unacceptability of (11)-(13) since GPs don't

6. The claim is that movement will never consume more than a bounded cost; the cost is independent of the length of the sentence. Some movement sentences may be easier than others (subject vs. object relatives). See [Church80] for more discussion.
7. In fact, the human processor may not be optimal. The functional argument observes that an optimal processor could process unbounded movement with bounded resources. This should encourage further investigation, but it alone is not sufficient evidence that the human processor has optimal properties.
8. Chomsky and Lasnik (personal communication) have each suggested that the competence model might generate a non-computable set. If this were indeed the case, it would seem unlikely that there could be a mapping onto the finite performance world.
9. *Acceptability* is a formal term; see footnote 3.
10. A push down automata (PDA) is a formalization of stack machines.

center-embed very deeply. Determinism offers an additional constraint on memory allocation which provides an account for the data.

(11) #The horse raced past the barn fell.
(12) #John lifted a hundred pound bags.
(13) #I told the boy the dog bit Sue would help him.

At first we believed the memory constraint alone would subsume Marcus' hypothesis as well as providing an explanation of the center-embedding phenomena. Since all FSMs have a deterministic realization,[11] it was originally supposed that the memory limitation guaranteed that the parser is deterministic (or equivalent to one that is). Although the argument is theoretically sound, it is mistaken.[12] The deterministic realization may have many more states than the corresponding non-deterministic FSM. These extra states would enable the machine to parse GPs by delaying the critical decision.[13] In spirit, Marcus' Determinism Hypothesis excludes encoding non-determinism by exploding the state space in this way. This amounts to an exponential reduction in the size of the state space, which is an interesting claim, not subsumed by FS (which only requires the state space to be finite).

By assumption, the garbage collection procedure must act "deterministically"; it cannot backup or undo previous decisions. Consequently, the machine will not only reject deeply center-embedded sentences but it will also reject sentences such as (14) where the heuristic garbage collector makes a mistake (takes a garden path).

(14) #Harold heard [that John told the teacher [that Bill said that Sam thought that Mike threw the first punch] yesterday].

YAP is essentially a stack machine parser like Marcus' Parsifal with the additional bound on stack depth. There will be a garbage collector to remove finished phrases from the stack so the space can be recycled. The garbage collector will have to decide when a phrase is finished (closed).

4. Closure Specifications

Assume that the stack depth should be correlated to the depth of center-embedding. It is up to the garbage collector to close phrases and remove them from the stack, so only center-embedded phrases will be left on the stack. The garbage collector could err in either of two directions; it could be overly ruthless, cleaning out a node (phrase) which will later turn out to be useful, or it could be overly conservative, allowing its limited memory to be congested with unnecessary information. In either case, the parser will run into trouble, finding the sentence unacceptable. We have defined the two types of errors below.

(15) Premature Closure: The garbage collector prematurely removes phrases that turn out to be necessary.

(16) Ineffective Closure: The garbage collector does not remove enough phrases, eventually overflowing the limited memory.

There are two garbage collection (closure) procedures mentioned in the psycholinguistic literature: Kimball's early closure [Kimball73, 75] and Frazier's late closure [Frazier79]. We will argue that Kimball's procedure is too ruthless, closing phrases too soon, whereas Frazier's procedure is too conservative, wasting memory. Admittedly it is easier to criticize than to offer constructive solutions. We will develop some tests for evaluating solutions, and then propose our own somewhat ad hoc compromise which should perform better than either of the two extremes, early closure and late closure, but it will hardly be the final word. The closure puzzle is extremely difficult, but also crucial to understanding the seemingly idiosyncratic parsing behavior that people exhibit.

5. Kimball's Early Closure

The bracketed interpretations of (17)-(19) are unacceptable even though they are grammatical. Presumably, the root matrix[14] was "closed off" before the final phrase, so that the alternative attachment was never considered.

(17) #Joe figured [that Susan wanted to take the train to New York] out.
(18) #I met [the boy whom Sam took to the park]'s friend.
(19) #The girl$_i$ applied for the jobs [that was attractive]$_i$.

Closure blocks high attachments in sentences like (17)-(19) by removing the root node from memory long before the last phrase is parsed. For example, it would close the root clause just before *that* in (21) and *who* in (22) because the nodes [$_{comp}$ that] and [$_{comp}$ who] are not immediate constituents of the root. And hence, it shouldn't be possible to attach anything directly to the root after *that* and *who*.[15]

(20) Kimball's Early Closure: A phrase is closed as soon as possible, i.e., unless the next node parsed is an immediate constituent of that phrase. [Kimball73]

(21) [$_S$ Tom said
[$_{S_-}$ that Bill had taken the cleaning out ...

(22) [$_S$ Joe looked the friend
[$_{S_-}$ who had smashed his new car ... up

11. A non-deterministic FSM with n states is equivalent to another deterministic FSM with 2^n states.
12. I am indebted to Ken Wexler for pointing this out.
13. The exploded states encode disjunctive alternatives. Intuitively, GPs suggest that it isn't possible to delay the critical decision; the machine has to decide which way to proceed.
14. A matrix is roughly equivalent to a phrase or a clause. A matrix is a frame with slots for a mother and several daughters. The root matrix is the highest clause.
15. Kimball's closure is premature in these examples since it is possible to interpret *yesterday* attaching high as in: *Tom said [that Bill had taken the cleaning out] yesterday.*

This model inherently assumes that memory is costly and presumably fairly limited. Otherwise, there wouldn't be a motivation for closing off phrases.

Although Kimball's strategy strongly supports our own position, it isn't completely correct. The general idea that phrases are unavailable is probably right, but the precise formulation makes an incorrect prediction. If the upper matrix is really closed off, then it shouldn't be possible to attach anything to it. Yet (23)-(24) form a minimal pair where the final constituent attaches low in one case, as Kimball would predict, but high in the other, thus providing a counter-example to Kimball's strategy.

(23) I called [the guy who smashed my brand new car up]. (low attachment)

(24) I called [the guy who smashed my brand new car] a rotten driver. (high attachment)

Kimball would probably not interpret his closure strategy as literally as we have. Unfortunately computer models are brutally literal. Although there is considerable content to Kimball's proposal (closing before memory overflows), the precise formulation has some flaws. We will reformulate the basic notion along with some ideas proposed by Frazier.

6. Frazier's Late Closure

Suppose that the upper matrix is not closed off, as Kimball suggested, but rather, temporarily out of view. Imagine that only the lowest matrix is available at any given moment, and that the higher matrices are stacked up. The decision then becomes whether to attach to the current matrix or to close it off, making the next higher matrix available. The strategy attaches as low as possible; it will attach high if all the lower attachments are impossible. Kimball's strategy, on the other hand, prevents higher attachments by closing off the higher matrices as soon as possible. In (23), according to Frazier's late closure, *up* can attach[16] to the lower matrix, so it does; whereas in (24), *a rotten driver* cannot attach low, so the lower matrix is closed off, allowing the next higher attachment. Frazier calls this strategy late closure because lower nodes (matrices) are closed as late as possible, after all the lower attachments have been tried. She contrasts her approach with Kimball's early closure, where the higher matrices are closed very early, before the lower matrices are done.[17]

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

Unfortunately, it seems that Frazier's late closure is too conservative, allowing nodes to remain open too long, congesting valuable stack space. Without any form of early closure, right branching structures such as (26) and (27) are a real problem; the machine will eventually fill up with unfinished matrices, unable to close anything because it hasn't reached the bottom right-most clause. Perhaps Kimball's suggestion is premature, but Frazier's is ineffective. Our compromise will augment Frazier's strategy to enable higher clauses to close earlier under marked conditions (which cover the right branching case).

(26) This is the dog that chased the cat that ran after the rat that ate the cheese that you left in the trap that Mary bought at the store that ...

(27) I consider every candidate likely to be considered capable of being considered somewhat less than honest toward the people who ...

Our argument is like all complexity arguments; it considers the limiting behavior as the number of clauses increase. Certainly there are numerous other factors which decide borderline cases (3-deep center-embedded clauses for example), some of which Frazier and Fodor have discussed. We have specifically avoided borderline cases because judgments are so difficult and variable; the limiting behavior is much sharper. In these limiting cases, though, there can be no doubt that memory limitations are relevant to parsing strategies. In particular, alternatives cannot explain why there are no acceptable sentences with 20 deep center-embedded clauses. The only reason is that memory is limited; see [Chomsky59a,b], [Bar-Hillel61] and [Langendoen75] for the mathematical argument.

7. A Compromise

After criticizing early closure for being too early and late closure for being too late, we promised that we would provide yet another "improvement". Our suggestion is similar to late closure, except that we allow one case of early closure (the A-over-A early closure principle), to clear out stack space in the right recursive case.[18] The A-over-A early closure principle is similar to Kimball's early closure principle except that it waits for two nodes, not just one. For example in (28), our principle would close $[_1$ that Bill said $S_2]$ just before the *that* in S_3 whereas Kimball's scheme would close it just before the *that* in S_2.

16. Deciding whether a node can or cannot attach is a difficult question which must be addressed. YAP uses the functional structure [Bresnan (to appear)] and the phrase structure rules. For now we will have to appeal to the reader's intuitions.

17. Frazier's strategy will attach to the lower matrix even when the final particle is required by the higher clause as in: *?I looked the guy who smashed my car up*, or *?Put the block which is on the box on the table.*

18. Early closure is similar to a compiler optimization called tail recursion, which converts right recursive expressions into iterative ones, thus optimizing stack usage. Compilers would perform the optimization only when the structure is known to be right recursive; the A-over-A closure principle is somewhat heuristic since the structure may turn out to be center-embedded.

(28) John said [$_1$ that Bill said [$_2$ that Sam said [$_3$ that Jack ...

(29) The A-over-A early closure principle: Given two phrases in the same category (noun phrase, verb phrase, clause, etc.), the higher closes when both are eligible for Kimball closure. That is, (1) both nodes are in the same category, (2) the next node parsed is not an immediate constituent of either phrase, and (3) the mother and all obligatory daughters have been attached to both nodes.

This principle, which is more aggressive than late closure, enables the parser to process unbounded right recursion within a bounded stack by constantly closing off. However, it is not nearly as ruthless as Kimball's early closure, because it waits for two nodes, not just one, which will hopefully alleviate the problems that Frazier observed with Kimball's strategy.

There are some questions about the borderline cases where judgments are extremely variable. Although the A-over-A closure principle makes very sharp distinctions, the borderline are often questionable.[19] See [Cowper76] for an amazing collection of subtle judgments that confound every proposal yet made. However, we think that the A-over-A notion is a step in the right direction; it has the desired limiting behavior, although the borderline cases are not yet understood. We are still experimenting with the YAP system, looking for a more complete solution to the closure puzzle.

In conclusion, we have argued that a memory limitation is critical to reducing performance model complexity. Although it is difficult to discover the exact memory allocation procedure, it seems that the closure phenomenon offers an interesting set of evidence. There are basically two extreme closure models in the literature, Kimball's early and Frazier's late closure. We have argued for a compromise position; Kimball's position is too restrictive (rejects too many sentences) and Frazier's position is too expensive (requires too much memory for right branching). We have proposed our own compromise, the A-over-A closure principle, which shares many advantages of both previous proposals without some of the attendant disadvantages. Our principle is not without its own problems; it seems that there is considerable work to be done.

By incorporating this compromise, YAP is able to cover a wider range of phenomena[20] than Parsifal while adhering to a finite state memory constraint. YAP provides empirical evidence that it is possible to build a FS performance device which approximates a more complicated competence model in the easy acceptable cases, but fails on certain unacceptable constructions such as closure violations and deeply center embedded sentences. In short, a finite state memory limitation simplifies the parsing task.

19. In particular, the A-over-A early closure principle does *not* account for preferences in sentences like: *I said that you did it yesterday* because there are only two clauses. Our principle only addresses the limiting cases. We believe there is another related mechanism (like Frazier's Minimal Attachment) to account for the preferred *low attachments*. See [Church80].

20. The A-over-A principle is useful for thinking about conjunction.

8. References

Bar-Hillel, Perles, M., and Shamir, E., *On Formal Properties of Simple Phrase Structure Grammars*, reprinted in *Readings in Mathematical Psychology*, 1961.

Chomsky, *Three models for the description of language*, I.R.E. Transactions on Information Theory, vol. IT-2, Proceedings of the symposium on information theory, 1956.

Chomsky, *On Certain Formal Properties of Grammars*, Information and Control, vol 2, pp. 137-167, 1959a.

Chomsky, *A Note on Phrase Structure Grammars*, Information and Control, vol 2, pp. 393-395, 1959b.

Chomsky, *On the Notion "Rule of Grammar"*, (1961), reprinted in J. Fodor and J. Katz, eds., pp 119-136, 1964.

Chomsky, *A Transformational Approach to Syntax*, in Fodor and Katz, eds., 1964.

Cowper, Elizabeth A., *Constraints on Sentence Complexity: A Model for Syntactic Processing*, PhD Thesis, Brown University, 1976.

Church, Kenneth W., *On Memory Limitations in Natural Language Processing*, Masters Thesis in progress, 1980.

Frazier, Lyn, *On Comprehending Sentences: Syntactic Parsing Strategies*, PhD Thesis, University of Massachusetts, Indiana University Linguistics Club, 1979.

Frazier, Lyn & Fodor, Janet D., *The Sausage machine: A New Two-Stage Parsing Model*, Cognition, 1979.

Kimball, John, *Seven Principles of Surface Structure Parsing in Natural Language*, Cognition 2:1, pp 15-47, 1973.

Kimball, *Predictive Analysis and Over-the-Top Parsing*, in *Syntax and Symantics IV*, Kimball editor, 1975.

Langendoen, *Finite-State Parsing of Phrase-Structure Languages and the Status of Readjustment Rules in Grammar*, Linguistic Inquiry Volume VI Number 4, Fall 1975.

Lasnik, H., *Remarks on Co-reference*, Linguistic Analysis, Volume 2, Number 1, 1976.

Marcus, Mitchell, *A Theory of Syntactic Recognition for Natural Language*, MIT Press, 1979.

Woods, William, *Transition Network Grammars for Natural Language Analysis*, CACM, Oct. 1970.