An Architecturally-based Theory of Human Sentence Comprehension

Richard L. Lewis
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
rick@cs.cmu.edu

Abstract

Real-time language comprehension is an important area of focus for a candidate unified theory of cognition. In his 1987 William James lectures, Allen Newell sketched the beginnings of a comprehension theory embedded in the Soar architecture. This theory, NL-Soar, has developed over the past few years into a detailed computational model that provides an account of a range of sentence-level phenomena: immediacy of interpretation, garden path effects, unproblematic ambiguities, parsing breakdown on difficult embeddings, acceptable embedding structures, and both modular and interactive ambiguity resolution effects. The theory goes beyond explaining just a few examples, it addresses over 80 different kinds of constructions. Soar is not merely an implementation language for the model, but plays a central theoretical role. The predictive power of NL-Soar derives largely from architectural mechanisms and principles that shape the comprehension capability so that it meets the real time constraint.

Introduction

In his 1987 William James lectures, Allen Newell sketched the beginnings of a comprehension theory embedded in the Soar architecture (Newell, 1990; Rosenbloom et al., 1993, this volume). This theory, NL-Soar, has developed over the past few years into a detailed computational model that provides an account of a range of important sentence-level phenomena: immediacy of interpretation, interactive and modular ambiguity resolution effects, garden path effects, unproblematic ambiguities, parsing breakdown on difficult embeddings and acceptable embedding structures. Thus, NL-Soar shares the goal of unified, broad coverage with a number of recent theories in psychology, linguistics, and computational linguistics (Just and Carpenter, 1987; Gibson, 1991; Pritchett, 1992; Jurafsky, 1992).

How should Soar comprehend language? A modular approach to mental architecture (e.g., (Fodor, 1983)) might suggest adding a separate language module to Soar. This approach is perfectly consistent with Soar, since the working memory can act as a bus that admits additional processing modules (Newell, 1990). However, by taking this approach, we miss the opportunity of discovering ways in which comprehension shares architectural mechanisms with the rest of cognition. The approach we have taken with NL-Soar (Lehman et al., 1991; Newell, 1990) is to construct a model that embeds comprehension *within* Soar. In the following sections, we examine how the architecture, along with the functional requirements of real-time comprehension, shape the NL-Soar model.

Immediacy of interpretation

Our subjective experience is that we comprehend language incrementally, understanding each word as it is heard or read. As a hypothesis about the comprehension process, this has been advanced as the principle of *immediacy of interpretation*, and it is supported by much experimental evidence. This immediacy requirement extends to syntactic, semantic, and referential processing (e.g., (Marslen-Wilson, 1975; Just and Carpenter, 1987)).

Real-time immediacy constrains NL-Soar because Soar as a cognitive theory specifies approximate time constants for architectural processes. The most rapid operators take ~50–100 ms (Newell, 1990)[1]. To attain reading or listening rates of 200-300 words per minute, NL-Soar must comprehend each word with just a few (~3-6) operators.

Figure 1 shows how the basic organization of NL-Soar responds to this constraint. In the top space, *comprehension operators* apply to the incoming words. These operators incrementally build up two structures in working memory: the *situation model*, representing the content

This research was sponsored by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government.

[1] This is a mapping of Soar architectural mechanisms onto human constants, not a statement about actual computer system run times.

of the discourse, and the *utterance model*, representing the syntactic structure of the utterance. To achieve recognitional comprehension, the multiple knowledge sources required to implement the operators must be directly available in Soar's recognition memory. In previous work, we have demonstrated that it is possible to *deliberately* implement these operators in lower problem spaces that represent syntactic, semantic, and referential knowledge. For example, the syntax space contains *link* operators that establish structural relations in the utterance model. Constraints on these operators are represented independently in still lower spaces. Chunking over this deliberate process produces operators that recognitionally apply the separate knowledge sources (Lehman et al., 1991). The computational benefits are real: over a corpus of 61 sentences (designed to test syntactic coverage, not tuned to maximize chunk transfer) NL-Soar moved from comprehending *no* words by recognition to comprehending ~80% of the words recognitionally (i.e., without impasse) (Steier et al., 1993). While this does not provide a theory of initial language acquisition, it does demonstrate that chunking is capable of producing comprehension operators that satisfy the real time constraint.

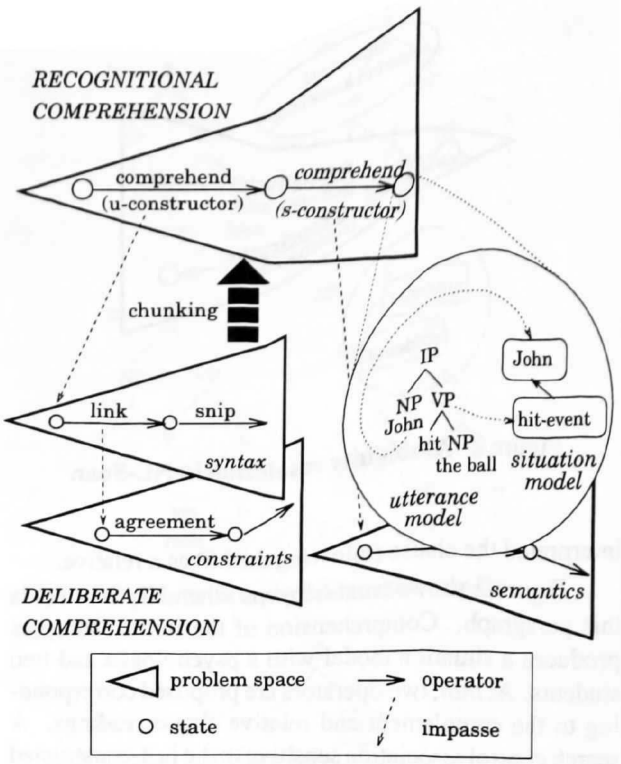

Figure 1: **Structure of NL-Soar.**

Ambiguity resolution: interactive and modular effects

A theory of comprehension must specify what knowledge is brought to bear in resolving local ambiguities, and how and when that knowledge is brought to bear. The nature of ambiguity resolution is at the heart of the modularity debate in sentence processing: is there an *autonomous syntactic parser* that operates without appeal to other knowledge sources, or is comprehension an *interactive* process in which multiple knowledge sources (including syntax) interact rapidly to produce the meaning?

The empirical results are mixed: a number of studies have demonstrated the effect of semantics (e.g., (Just and Carpenter, 1992)) and context (e.g., (Tyler and Marslen-Wilson, 1977), and a number of studies have demonstrated the lack of such effects (e.g., (Ferreira and Clifton, 1986)). Theories that account for both modular and interactive effects are just beginning to emerge (Just and Carpenter, 1992; Britt et al., 1992; Holbrook et al., 1992).

Before examining ambiguity in NL-Soar, consider the structure of comprehension operators in somewhat more detail. Figure 1 shows that comprehension operators fall into separate classes: *u-constructors* build up the utterance model (the syntactic structure of the utterance), and *s-constructors* build up the situation model (the semantic content of the utterance). (Not shown are the *referential* operators which perform reference resolution.)[2]

How does ambiguity arise? Choice points arise in problem spaces when multiple operators are applicable at a given state. Thus, syntactic ambiguity in NL-Soar arises when multiple u-constructors (corresponding to different syntactic paths) propose themselves. This generation of multiple alternatives occurs in parallel, since associations in the recognition memory fire in parallel. The *selection* of the appropriate operator may now be effected by search control associations that encode semantic and contextual knowledge. As an example, consider how NL-Soar might model the Crain and Steedman (1985) experiment, which demonstrated that referential context may affect the resolution of certain ambiguities. The following paragraph (adapted from the original material) illustrates:

> A psychologist was counseling two students. The psychologist argued with one student. The other student remained quiet. The psychologist told the student that he argued with that the horse raced past the barn.

The *that* in the final sentence introduces a clause which could be taken as the complement of *told* or a restrictive relative clause modifying *student*. When the NP identified more than one referent (as *student* does), subjects

[2]Decomposing the knowledge this way across different operators (as opposed to a single fully integrated comprehension operator as in earlier versions of NL-Soar) increases the generality of the resulting proposal rules, which should increase the asymptotic efficiency of the system. Further data from the implemented system is required to fully explore this issue.

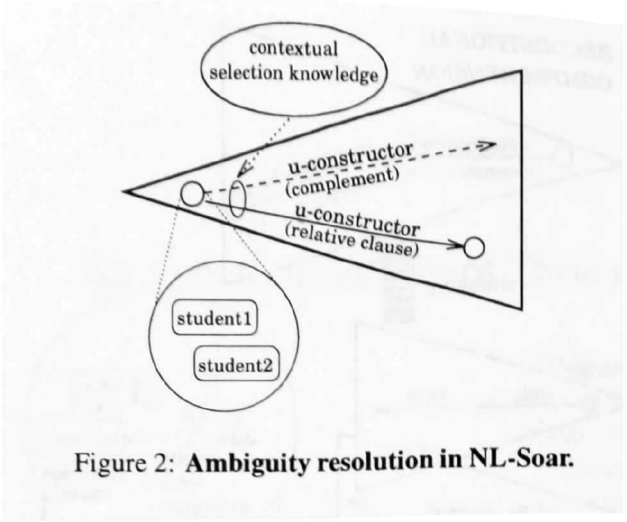

Figure 2: **Ambiguity resolution in NL-Soar.**

interpreted the clause following the NP as a relative.

Figure 2 shows what happens when NL-Soar reads this paragraph. Comprehension of the initial sentences produces a situation model with a psychologist and two students. At *that*, two operators are proposed corresponding to the complement and relative clause readings. A search control association sensitive to the just-constructed model—that there are two students—then guides the interpretation down the correct path.

Such beneficial effects of context and semantics depend on having the appropriate search control associations immediately available, but there is nothing in the architecture of NL-Soar that guarantees this will be the case. Indeed, if the relevant associations have not yet been learned, then NL-Soar may yield classic modularity effects since it cannot bring to bear all the appropriate knowledge sources in real-time.

The set of u-constructors exhibits many characteristics of an autonomous syntax module (Lewis, 1993). Thus, NL-Soar has much in common with theories proposing a syntax module that generates alternative structures in parallel, arbitrated by semantic/contextual knowledge sources (e.g., (Warner and Glass, 1987)). In NL-Soar, however, the parallelism and fine-grained control arise directly from Soar's recognition memory and control structure. NL-Soar also makes novel qualitative predictions about the potential effect of *learning*. The more novel the semantic content and context for an utterance, the more likely modular effects will arise. The corollary prediction is that modularity effects can be reduced with the right kind of experience.

Garden path effects and unproblematic ambiguities

A *garden path* effect arises when a reader or listener attempts to comprehend a grammatical sentence with a local ambiguity, misinterprets the ambiguity, and is unable to recognitionally recover the correct interpretation. The result is an impression that the sentence is ungrammatical. The classic example (1) is due to Bever (1970):

(1) #The horse raced past the barn fell.

The subjective experience provides compelling linguistic evidence for the difficulty of these sentences, but additional evidence comes from reading times and grammaticality judgments (e.g., (Warner and Glass, 1987)). The reduced relative construction in (1) is but one kind of garden path; Figure 3 provides a sample of a collection of over 25 different types (see also (Gibson, 1991; Pritchett, 1992; Lewis, 1992)).

NP-modifier/relative: *The Russian women loved died.*

Short reduced relative: *The boat floated sank.*

Object/subject specifier: *I convinced her professors hate me.*

Object/object: *Sue gave the man racing the car.*

Prep object/verb object (German): *daß der Entdecker von Amerika erst im 18 Jahrhundert erfahren hat* ("that the discoverer originally learned of America in the 18th century")

Figure 3: **Some garden path constructions.**

NP/NP specifier: *Without her we failed. Without her contributions we failed.*

Noun/adjective: *The square is red. The square table is red.*

Double object: *I gave her earrings. I gave her earrings to Sally.*

Long distance gaps: *Who do you believe? Who do you believe John suspects Steve knows Bill hates?*

Multiple compounding: *We admire their intelligence. We admire their intelligence agency policy decisions.*

Figure 4: **Some unproblematic ambiguities.**

Unproblematic ambiguities provide data that complements the garden path constructions. Consider the pair of sentences in (2):

(2) (a) I know John very well.
(b) I know John is nice.

There is a local ambiguity at *John*, since it could be the direct object of *know* or the subject of an incoming clause. Regardless of the final outcome, the sentence causes no perceptible processing difficulty. Figure 4 provides a sample of a collection of over 30 unproblematic ambiguities.

Since NL-Soar is a single path comprehender, there must be some capability for reanalysis in case a wrong path is taken. The reanalysis mechanism must work satisfy several constraints. 1) It must be powerful enough to handle the range of unproblematic ambiguities, but not

so powerful that it fails to predict the garden path effects. 2) It must work with the given inconsistent syntactic state (there are no previous states to backtrack to). 3) It must be real-time (a part of recognitional comprehension) 4) It must work without reprocessing the input (Lewis, 1992).

NL-Soar's reanalysis mechanism is *simple destructive repair*. It consists of a single operator, *snip*, that breaks a syntactic link in the utterance model. *Snip* exists in the implementation space for u-constructors along with the *link* operators (Figure 1). Through chunking, the reanalysis process becomes part of the top-level comprehension operators, yielding *recognitional repair*.

Proposing a snip for every syntactic relation in the utterance model would lead to a large set of operators in working memory. Such undiscriminated sets permit the generation of exponential cross products in the recognition match (Tambe et al., 1990). This *expensive chunk* problem is a non-trivial effect observed in implemented Soar systems, including early versions of NL-Soar. The resulting slowdown compromises a basic assumption of Soar that the recognition match is an efficient process. This jeopardizes the ability of the model to satisfy the real-time immediacy constraint.

To control this overgeneration, snips are only proposed for relations *local*[3] to a node where a problem has been detected. For example, consider the repair process for (2b) in Figure 5. Part (a) of the figure shows the syntactic structure after comprehending *I know John*: *John* is in the complement position of *know*[4]. Link operators are permitted to assign constituents to structural positions regardless of whether the positions are occupied or not (as long as the link is grammatical). Thus, when *is* arrives, it is projected to a sentential phrase and linked to complement position of *know* (b). An inconsistency is detected at the VP node (boxed in the figure): two constituents occupying the same structural position. A snip operator is proposed to break the duplicate link, which is local to the VP. This releases the NP *John* (c), which is then attached as the subject of *is*, completing the repair (d).

Figure 6 shows the structure that results from processing a subject/object ambiguity which, unlike (2), *does* cause processing difficulty:

(3) #Since Jay always jogs a mile seems light work.

In this case, the fronted clause *Since Jay always jogs a mile* is adjoined to the S projected from *seems*, but processing then reaches a dead end. The appropriate *snip* (removing *mile*) is not generated since it is not local to the detected problem (the S with the missing subject).

In addition to duplicate relations, two other inconsistencies may generate snips: missing obligatory constituents, and attachment to competing syntactic senses of the same lexical item. The latter permits the repair mechanism to extend in a simple way to cover reanalysis involving lexical categorial ambiguity (e.g., the *noun/adjective* ambiguity in Figure 4).

[3]More precisely, local to the maximal projection containing the node.

[4]This figure is a simplification of the actual phrase structure used in NL-Soar, which corresponds to X-Bar syntax (Chomsky, 1986).

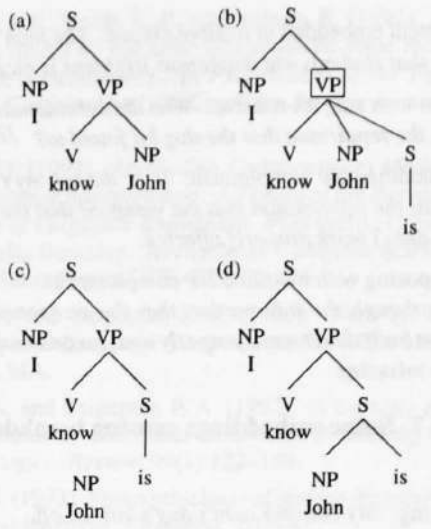

Figure 5: **Simple destructive repair.**

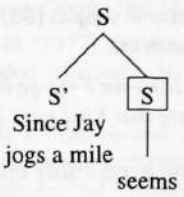

Figure 6: **Failure to repair.**

NL-Soar's simple repair mechanism accounts for over 50 garden path constructions and unproblematic ambiguities (including all of the examples listed in Tables 3 and 4)[5]. These predictions were primarily derived by hand-simulation; implementation of the new system is underway.

Parsing breakdown and acceptable embeddings

Some constructions *without* structural ambiguity are difficult to comprehend. Consider the center-embedded sentence (4):

(4) #The man that the woman that the dog bit likes eats fish.

[5]The best-known GP construction unaccounted for is the argument/adjunct ambiguity: *#The patient persuaded the doctor that he was having trouble with to leave.* The phenomena surrounding argument/adjunct ambiguities are fairly complex; these constructions will be an area for future research.

NP complement embedded in relative clause: *The man who the possibility that students are dangerous frightens is nice.*

Wh-question with subject-relative: *Who did John donate the furniture that the repairman that the dog bit found to?*

Cleft with modified NP complement: *It is the enemy's defense strategy that the information that the weapons that the government built didn't work properly affected.*

Though-preposing with modified NP complements: *Surprising though the information that the weapons that the government built didn't work properly was, no one took advantage of the mistakes.*

Figure 7: **Some embeddings causing breakdown.**

Left-branching: *My cousin's aunt's dog's tail fell off.*

Pseudo-cleft with relative: *What the woman that John married likes is smoked salmon.*

Post-verbal untensed sentential subject (SS): *I believe that for John to smoke would annoy me.*

4-NP initial (Japanese): *John-wa Fred-ga biiru-o Dave-ni ageta koto o kiita. ("John heard that Fred gave beer to Dave.")*[6]

Figure 8: **Some acceptable embeddings.**

In general, people have trouble beyond one level of embedding. This difficulty has been empirically verified with grammaticality judgment and comprehension tasks (e.g., (Larkin and Burns, 1977)). There are a variety of similar constructions causing breakdown; Gibson (1991) presents the most complete set. Figure 7 presents some samples.

Not all multiple embeddings cause difficulty. For example, right-branching may occur without bound (Kimball, 1973):

(5) The dog saw the cat which chased the mouse into the house that Jack built.

Figure 8 presents a sample of a corpus of over 25 acceptable embeddings. Such constructions constrain theories of parsing breakdown in the same way that the unproblematic ambiguities constrain garden path theories.

In order to see how NL-Soar accounts for these constructions, we must first examine in more detail the representation of the utterance model in working memory. Soar's working memory consists of attribute-value structures. The partial utterance model is represented by attribute-value pairs that index words and constituents by their potential syntactic relationships. For example, the words *the dog* might first appear in working memory as:[7]

[6]Thanks to Brad Pritchett for this example.

[7]The relations NL-Soar actually uses correspond to X-Bar positions.

```
Assigns:   ^spec dog
Receives:  ^spec the ^subj dog ^obj dog
```

This means that *dog* can assign a specifier role and receive an object or subject role, and *the* can receive a specifier role. As processing continues, additional constituents can be added to each attribute. Parsing is a bottom-up process that consists of matching potential assigners and receivers and establishing the links permitted by grammatical constraints.

Soar's attribute-value representation permits the creation of large undiscriminated sets in working memory: a single attribute may index many values. As noted above, this can lead to exponential slowdown in the recognition match. To avoid these combinatorics, NL-Soar restricts each attribute to having just *two* associated values. Working memory capacity for syntax thus emerges from an interaction of the partial construction, which determine the available syntactic discriminators, and a limit on how much material each discriminator may index.

Figure 9 shows how this restriction predicts breakdown on (4). The breakdown arises because one syntactic attribute (the subject attribute) must index *three* constituents: the NPs *man*, *boy*, and *dog*. When *dog* is comprehended, one of the earlier NPs must be removed. Thus, all of the subject NPs will not be available for attachment when the verbs finally arrive. By contrast, the acceptable right branching structure (5) can be handled because only one NP must be available for modifier attachment at any given point in the sentence. NL-Soar accounts for over 30 difficult and acceptable embeddings, including all of the constructions in Tables 7 and 8[8]. These constructions include several interesting phenomena in head-final languages.

[8]A number of unacceptable constructions involving sentential subjects may be ruled out for independent grammatical reasons (Koster, 1978).

Read *"the man"*
```
Assigns:   ^modifier man
Receives:  ^subject man ^object man
```

Read *"the woman"*
```
A: ^modifier man woman
R: ^subject man woman ^object man woman
```

Read *"the dog"*
```
A: ^modifier dog woman
R: ^subject dog woman ^object dog woman
```

Figure 9: **Breakdown on center-embedding.**

Conclusion

NL-Soar is a computational model of sentence comprehension that accounts for a broad range of important sentence-level phenomena, providing detailed predictions on garden path constructions, unproblematic ambiguities, difficult embeddings, and acceptable embeddings.

The characteristics of the model either derive directly from architectural mechanisms in Soar (conversion from deliberate to recognitional comprehension, parallel generation of structural alternatives at ambiguous points, fine-grain control over ambiguity resolution) or from the application of architectural principles to ensure that comprehension meets the real-time constraint (small set of top-level comprehension operators, controlled generation of the repair operator, limits on multiple-valued attributes in working memory).

Perhaps it comes as a surprise that building a comprehension model within a general cognitive architecture would prove fruitful. But as Allen said, *"There are more things in an architecture, Horatio, than are dreamt of in your theorizing."* (Newell, 1990).

Acknowledgments

This work would not have been possible without the collaborative efforts and invaluable guidance of Jill Lehman and the late Allen Newell. Thanks also to Brad Pritchett for help with syntactic theory.

References

Bever, T. G. (1970). The cognitive basis for linguistic structures. In Hayes, J. R., editor, *Cognition and the Development of Language*. Wiley, New York.

Britt, M. A., Perfetti, C. A., Garrod, S., and Rayner, K. (1992). Parsing in discourse: Context effects and their limits. *Journal of Memory and Language*, 31:293–314.

Chomsky, N. (1986). *Barriers*. MIT Press, Cambridge, MA.

Crain, S. and Steedman, M. (1985). On not being led up the garden path: The use of context by the psychological syntax processor. In Dowty, D. R., Karttunen, L., and Zwicky, A. M., editors, *Natural Language Parsing*. Cambridge University Press, Cambridge, U.K.

Ferreira, F. and Clifton, Jr., C. (1986). The independence of syntactic processing. *Journal of Memory and Language*, 25:348–368.

Fodor, J. A. (1983). *Modularity of Mind: An essay on faculty psychology*. MIT Press, Cambridge, MA.

Gibson, E. A. F. (1991). *A Computational Theory of Human Linguistic Processing: Memory Limitations and Processing Breakdown*. PhD thesis, Carnegie Mellon. Available as Center for Machine Translation technical report CMU-CMT-91-125.

Holbrook, J. K., Eiselt, K. P., and Mahesh, K. (1992). A unified process model of syntactic and semantic error recovery in sentence understanding. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 195–200.

Jurafsky, D. (1992). *An On-line Computational Model of Human Sentence Interpretation: A Theory of the Representation and use of Linguistic Knowledge*. PhD thesis, University of California, Berkeley. Available as Computer Science Division technical report UCB-CSD-92-676.

Just, M. A. and Carpenter, P. A. (1987). *The Psychology of Reading and Language Comprehension*. Allyn and Bacon, Boston, MA.

Just, M. A. and Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, 99(1):122–149.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, 2:15–47.

Koster, J. (1978). Why subject sentences don't exist. In Keyser, S. J., editor, *Recent Transformational Studies in European Languages*. MIT Press, Cambridge, MA.

Larkin, W. and Burns, D. (1977). Sentence comprehension and memory for embedded structure. *Memory and Cognition*, 5:17–22.

Lehman, J. F., Lewis, R. L., and Newell, A. (1991). Integrating knowledge sources in language comprehension. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 461–466.

Lewis, R. L. (1992). Recent developments in the NL-Soar garden path theory. Technical Report CMU-CS-92-141, School of Computer Science, Carnegie Mellon University.

Lewis, R. L. (1993). Architecture Matters: What Soar has to say about modularity. In Steier, D. and Mitchell, T., editors, *Mind Matters: Contributions to Cognitive and Computer Science in Honor of Allen Newell*. Erlbaum, Hillsdale, NJ. To appear.

Marslen-Wilson, W. D. (1975). Sentence perception as an interactive parallel process. *Science*, 189:226–227.

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts.

Pritchett, B. L. (1992). *Grammatical Competence and Parsing Performance*. University of Chicago Press, Chicago. In press.

Rosenbloom, P. N., Lehman, J. F., and Laird, J. E. (1993). Overview of soar as a unified theory of cognition: Spring 1993. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, Boulder, Colorado.

Steier, D. M., Lewis, R. L., Lehman, J. F., and Zacherl, A. L. (1993). Combining multiple sources of knowledge in an integrated intelligent system. *IEEE Expert*. To appear.

Tambe, M., Newell, A., and Rosenbloom, P. S. (1990). The problem of expensive chunks and its solution by restricting expressiveness. *Machine Learning*, 5:299–348.

Tyler, L. K. and Marslen-Wilson, W. (1977). The on-line effects of semantic context on syntactic processing. *Journal of Verbal Learning and Verbal Behavior*, 16:683–692.

Warner, J. and Glass, A. L. (1987). Context and distance-to-disambiguation effects in ambiguity resolution: Evidence from grammaticality judgments of garden path sentences. *Journal of Memory and Language*, 26:714–738.