Specifying Architectures for Language Processing: Process, Control, and Memory in Parsing and Interpretation

Richard L. Lewis
Department of Computer and Information Science
The Ohio State University
2015 Neil Avenue Mall
Columbus, OH 43210
rick@cis.ohio-state.edu

September 12, 1997

To appear in: Crocker, M., Pickering, M. and Clifton, C. Jr. editors, *Architectures and Mechanisms for Language Processing.* Cambridge University Press.

1 Introduction

An important goal in psycholinguistics is uncovering the *architecture* of human sentence comprehension. Most of the important issues in the field of sentence processing are questions about some aspect of the underlying architecture—for example, the relation of grammar and parser (Chomsky, 1965; Chomsky, 1980; Miller, 1962; Stabler, 1991), the modularity of syntactic processing (Ferreira & Clifton, 1986; Fodor, 1983; Forster, 1979; Frazier & Clifton, 1996; Just & Carpenter, 1992; Mitchell et al., 1992; Rayner et al., 1992), the number of interpretations pursued in parallel (Clark & Clark, 1977; Kurtzman, 1985; Gibson, 1991; Gorrell, 1987; MacKay, 1966), or the relation of linguistic processing to central cognition (Forster, 1979). The first goal of this paper is to define precisely what it means to have a functionally complete sentence processing architecture. The view of architecture developed here draws on general work on architecture in cognitive science (Anderson, 1983; Newell, 1973; Newell & Simon, 1972; Newell, 1980a; Newell, 1990; Pylyshyn, 1984), rather than focusing exclusively on the issue of modularity that has dominated discussions in sentence processing. This architectural analysis defines a set of functional constraints on theories, sharpens the set of questions that need to be answered, and reveals that some common theoretical approaches, including modularity, are incomplete in significant ways. The claim is that theories of human sentence processing should take the form of complete computational architectures.

A useful approach for discerning the shape of the human architecture is to look for extreme behavioral data points—phenomena that help sharply define both the impressive functional capacities of human comprehension as well as its limits. Taken together, these capacities and limitations comprise a bundle of behavioral oppositions; some of the major ones are summarized in Figure 1.

The second goal of this paper is bring to bear these functional and empirical constraints in proposing an architectural theory that embodies a simple set of claims about the processes, memories, and control structure underlying parsing and interpretation. The theory, NL-Soar, posits that human sentence processing is single path with a limited repair capa-

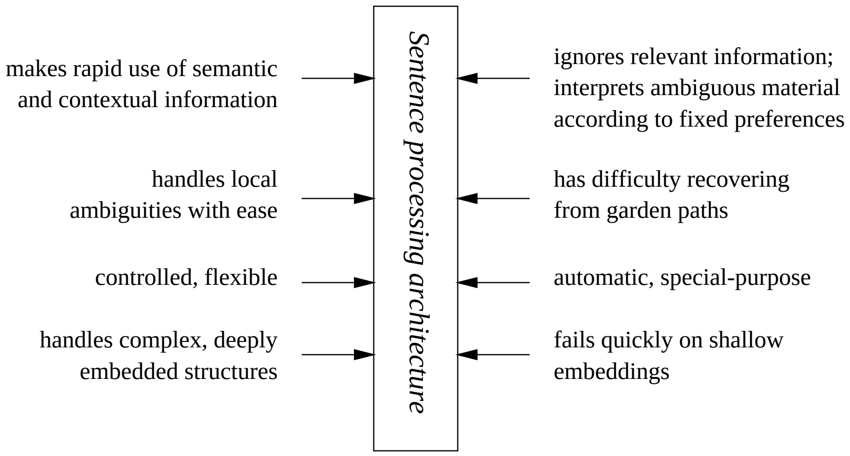

FIGURE 1: Human sentence comprehension can be characterized as a set of behavioral oppositions. These oppositions serve as useful empirical constraints by pushing on architectural models from opposite poles of capabilities and limitations. §2 describes the phenomena underlying these oppositions.

bility, has an automatic but flexible control structure that is open to effects of learning, and depends on a minimal short-term memory that gives rise to syntactic interference effects. The theory explains the behavioral puzzles in Figure 1. It explains why sentence processing can appear to ignore relevant information as it pushes down the garden path, yet at times be sensitive to semantic and pragmatic context. It explains why some misinterpretations are difficult to recover from, yet others are imperceptibly easy. It further explains how automatic processes can be redeployed in a controlled manner to recover from these difficult misinterpretations. Finally, it explains why human comprehension easily handles complex syntactic structures most of the time, yet fails miserably which pushed beyond certain very modest limits. Many of the novel aspects of the theory derive in part from the model's grounding in an independent theory of human cognitive architecture, Soar (Lehman et al., 1996; Newell, 1990; Newell, 1992).

The remainder of the paper is organized as follows: §2 provides a brief overview of the empirical constraints outlined in Figure 1. §3 describes a set of functional constraints by answering the question: What constitutes a theory of sentence processing architecture?

§4 presents a specific architectural theory that responds to the functional and empirical constraints. §5 describes the empirical implications in detail, and §6 concludes with a summary discussion, and some remarks on the general approach.

2 Opposing behavioral constraints on architecture

The first behavioral opposition in Figure 1 concerns the nature of ambiguity resolution: What guides the on-line interpretation of ambiguous material? A single example will serve to illustrate the major issues:

(1) The car examined by the mechanic was damaged.

Examined may be initially taken as the main verb, or as a reduced relative clause modifying *car*. The inanimacy of *car* makes it more likely that something was examining the car (relative clause reading) than that the car was examining something (main verb reading). The question is whether such semantic information can be used on-line to guide the human parser down the right path. Ferreira & Clifton (1986) presented reading time data that suggests that people ignore such non-syntactic information. On the other hand, Tanenhaus, Carlson, and Trueswell (1989) and Just and Carpenter (1992) present on-line studies suggesting that subjects *do* make rapid use of this semantic information in resolving the local ambiguity—they prefer the relative clause reading over the main verb reading when the initial NP is inanimate.

There is by now a large body of empirical work on both sides of the issue, some work showing that semantic or contextual information is ignored in first-pass reading, and some showing clear effects. The studies cover a range of information types and ambiguity types. Many theoretical approaches now acknowledge the need to accommodate, at some level, both kinds of effects—attributing the difference to differences in lexical/syntactic frequency, for example (MacDonald et al., 1994). But the data is perhaps more insidious than is often acknowledged. Just & Carpenter (1992) provided evidence for *individual differences* in use of semantic information. In other words, some subjects appeared modular, and some did not. The experiments used the same material as the Ferreira & Clifton (1986) study, across subjects. The implication for an architectural theory should be clear. Any architectural model must embody this opposition: it must be capable in principle of demonstrating both kinds of effects, and furthermore should identify the locus of variation that accounts for the individual differences (Just & Carpenter attribute them to working memory differences).

The remaining three oppositions in Figure 1 have received considerably less attention than the first. The second opposition concerns the the other half of the ambiguity resolution story: How do people revise their interpretations of ambiguous material based on later disambiguating information? An important source of data that can be used to reveal the nature of revision processes concerns the contrast between ambiguous structures that can give rise to noticeable garden path effects, such as the subject/object ambiguity in (2) (Frazier & Rayner, 1982), and those ambiguous structures that cause little or no perceptible difficulty, such as the subject/object ambiguity in (3) (Juliano & Tanenhaus, 1994; Mitchell, 1987; Pritchett, 1988):

(2) Although Mary forgot her husband didn't seem very upset yesterday.

(3) a. Mary forgot her husband at the airport yesterday.

b. Mary forgot her husband needed a ride yesterday.

The structure in (2) can give rise to an impression of ungrammaticality, indicating a failure of reanalysis. The structure in (3) may produce longer reading times in the disambiguating region (e.g., an extra 50 ms on *needed* compared to unambiguous structures with overt complementizers (Ferreira & Henderson, 1990)), but does not usually cause noticeable difficulty, regardless of which interpretation is required. Why does reanalysis fail in structures such as (2), and succeed in structures such as (3)? A theory of reanalysis is required to explain the contrasts (Frazier & Rayner, 1982)—ambiguity resolution principles alone are insufficient.

The third opposition pits the apparent automaticity of parsing and interpretation, against the capacity for more controlled, flexible comprehension. As Fodor et al. (1975) noted, the "...overwhelmingly puzzling aspect of sentence comprehension is how people manage to do it so quickly." This impressive rapidity combined with the complexity of the task, the modular effects cited earlier, and certain other theoretical considerations have led many to posit automatic, informationally encapsulated processes underlying parsing and interpretation (Laberge & Samuels, 1974; Fodor, 1983; Frazier, 1978; Forster, 1979).

But the fact is that people *do* recover from garden path effects, such as those discussed above, by some strategy of rereading, or more careful comprehension[1]. Eye-tracking studies also reveal frequent and highly selective regressions during reading, to reparse or reconsider some part of the passage (Frazier & Rayner, 1982). Somehow, the automatic processes that initially delivered the wrong output must be overridden or redirected in some way. Furthermore, subjects are able to adopt quite different strategies for comprehension, ranging from extremely rapid surface skimming to directed comprehension in the service of specific problem solving goals (Just & Carpenter, 1987). How can we reconcile this flexibility in comprehension with automaticity? Frazier (1990) discussed a form of this problem with respect to reanalysis, and pointed out that simply assuming post-first-pass reanalysis is something accomplished by the "central processor" leads to some undesirable consequences, including the duplication of processes and knowledge in the modular linguistic processor and the central processor. In short, we cannot leave unexplicated the relation between automatic linguistic processes and more controlled central processes (Forster, 1979).

The fourth opposition concerns the contrast between easily-parsed, complex embedded structures, and relatively simple embeddings that cause complete breakdown. Consider the structure in (4), which contains four embedded clauses:

(4) The bird chased the mouse that scared the cat that saw the dog that ate the pumpkin that grew in the garden.

Such right-branching structures can be embedded essentially without limit. But one of the first clearly identified psycholinguistic phenomenon was difficulty on center-embedded structures, which can cause severe problems at just two embeddings (Miller & Chomsky,

[1] Or in some severe cases, by explicit instruction—but even here the point is the same, if the subject comes to perceive the sentence as grammatical.

1963; Miller & Isard, 1964):

(5) The salmon that the man that the dog chased smoked fell.

The story is considerably more complex than this (Cowper, 1976; Gibson, 1991; Lewis, 1997b). For example, double center-embeddings are not always unacceptable (which rules out a simple appeal to ungrammaticality). Consider the following subject sentence construction (Cowper, 1976):

(6) That the food that John ordered tasted good pleased him.

In (6), two sentences (*the food tasted good, John ordered*) are center-embedded within the matrix clause (*That... pleased him.*) These effects are not due to ambiguity—they appear to be problems with a limited short term memory (Miller & Isard, 1964). Thus, an architectural theory must specify the memory structure underlying parsing that accounts for acceptable embeddings, including acceptable center-embeddings, and unacceptably difficult structures.

3 What is a sentence processing architecture?

Before considering issues in specifying architectures for sentence processing, it is important to understand what a computational architecture is generally. The idea of an architecture was imported into cognitive psychology from computer science (Bell & Newell, 1971; Newell, 1973; Newell, 1990; Pylyshyn, 1984), and computer architectures provide a useful starting point for our purposes.

3.1 Functional invariants: Process, memory and control

In computer systems, the architecture of a computer refers to the *fixed* hardware structure that supports computation. The fixed structure of any digital computer—whether PC, mainframe, parallel or single-processor—can be described in terms of functional units that store and process bit vectors. The description is cast in a *register-transfer language* (e.g., Bell & Newell, 1971). Expressions in such a language are composed of registers (bit vectors),

$$P : A \leftarrow A + B$$

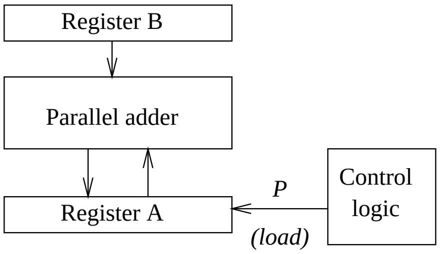

FIGURE 2: Part of a register transfer description of a digital computer. Expressions in the register transfer language consist of primitive *processes* on some registers (*memories*), preceded by a *control* prefix indicating what triggers the given processes. The block diagram indicates the functional logic units for implementing the given expression, which adds registers A and B and transfers the result to A whenever control bit P is set.

operations on registers, and control signals for triggering operations. For example, the expression in Figure 2 says that when bit P is set, then the result of adding register A and B is transferred to register A. The left hand part of the expression is like a conditional statement that indicates when the operation should take place. A set of such expressions can provide a complete functional specification of the architecture of a computer, which can then be implemented as a collection of logic circuits.

There are three important functional invariants that hold across all computational architectures and implementations. The hypothesis is that these invariants hold not only of digital computers, but any natural, artificial, or theoretical device that performs computations, including classic symbolic architectures, connectionist networks, and brains.

> **All computational architectures must support, minimally, the following functions: (1) a set of processing primitives; (2) a memory; and (3) a control structure, which specifies how the computational processes unfold over time.**

Support for these functional invariants comes from both empirical practice in computer science, and theoretical analysis. All physically constructed computational devices embody these minimal functions, as do all theoretical models (e.g., Turing machines, Post productions, RAMs; Minsky, 1967) developed for investigating computability. Newell (1980a) also emphasized this tripartite functional division in his description of physical symbol systems, which are general computational architectures.

Each of these functions can be seen clearly in the register transfer description of computer architectures. The *processing primitives* consist of the loads, adds, shifts, etc. that provide the building blocks for composing more complex operations. The *memory* at the register transfer level consists of registers which store bit vectors, or banks of registers operating together as larger units addressed by position (i.e., RAMs). The *control structure* is specified by the set of control prefixes for each expression, which dictate exactly when each primitive process occurs. Any number of such processes may occur simultaneously; at the register transfer level, parallelism is the norm for digital computers.

Another important property of the register transfer functional description is that it abstracts away from the particular set of logical circuits chosen to implement the architecture (e.g., how the AND/OR gates are arranged), which in turn abstracts away from properties of the electrical circuits (e.g., their voltage and resistance properties). In general, physical computational systems can be described as a hierarchy of systems levels. Each level is a functional description that abstracts away from implementation details of the lower levels.

Computers also illustrate another important principle: all computational architectures distinguish fixed structure from variable content, and system behavior depends on *both*. We can capture this in the following slogan equation:

Architecture (fixed structure) + content (variable) $\Longrightarrow$ behavior[2]

Newell (1990) points out that for many systems, including natural cognitive architectures, we should speak of "relatively fixed" structure to allow for architectures that change or develop (perhaps slowly) over time.

What is the relevance of all this for psycholinguistics? If we take sentence processing to be an information processing task, i.e. one requiring computation, then we can assume that underlying this computation is some kind of fixed architecture that realizes the three

[2]In a programmable architecture, the variable content includes the program, which is the primary determinant of behavior. Not all architectures are programmable, but all architectures process variable content.

basic functions outlined above. Furthermore, we can assume that this architecture can and should be described at multiple levels, each involving a different technology. This in no way asserts that the architecture of human sentence comprehension will be anything like a digital computer or a Turing machine. Discerning the right shape of the functional architecture is the task of empirical psycholinguistics:

> **An architecture for sentence processing is the fixed computational structure that supports comprehension—the control structures, memories, and primitive processes for parsing and interpretation.**

Such a definition may seem to be too general to have any import for theory development, but this is not the case. The following sections describe in detail the implications of this definition for models of human sentence processing.

3.2 "Module geography": Why modules are not enough

It is worthwhile to compare this view of sentence processing architecture with the view that has dominated psycholinguistics since the late 1970s. The dominant view holds that uncovering the architecture of sentence processing means identifying the independent processing modules that comprise comprehension. But the definition of architecture above gives no special mention of modules. The reason is simple: there is no a priori *functional* requirement for separate processing modules, as there is for the functions of memory, process, and control. The decomposition of architecture into modules is perfectly compatible with this view of architecture, but not required by it. Modularity is certainly an important empirical issue (see, for example, the references cited earlier) but it is not the only one.

In fact, just laying out a set of processing modules and their connections—what J.D. Fodor (1990) dubbed "module geography"—does not specify a functionally complete computational architecture. Perhaps most significantly, it does not specify any structure at all for the memories used in processing. For example, it does not specify the short-term memory for partial structures, or the long-term memory of syntactic or lexical knowledge. It also does not specify the primitive processes that the modules use. It does not specify control of processing within the modules. Finally, it does not specify the relation of the control of central processing to the control of the modular processes. To be clear: the theoretical claims that modular theories make are important ones which often yield testable predictions. But they are a necessary, not sufficient, part of the characterization of sentence processing architecture.

Abstracting away from irrelevant details is a theoretical virtue, but the kinds of abstractions that module geography makes can lead to incorrect inferences from data. That such a possibility exists is clearly demonstrated by the working memory research of Just & Carpenter (1992). Briefly, Just and Carpenter have argued that some garden path effects that were previously interpreted in terms of a syntactically encapsulated module can instead be explained by individual differences in working memory capacity. Such an explanation is not considered in a theoretical framework that systematically ignores the role of memory structures in parsing. This point should be taken regardless of whether one is convinced by the current body of empirical support for this particular model—the fact remains that such an explanation *could* in principle account for the data, and these alternative explanations are only discovered by developing functionally complete architectures. The next few sections describes what it means to specify such an architecture.

3.3 Specifying processes

We can characterize processes in two important ways. First, processes can be described by the way they change the content of memory; for example, making syntactically legal attachments in a parse tree held in a short-term memory. Second, the chronometric properties of process can be specified: how much time they require, and how their duration varies as a function of other variables. All of these specifications can be made at varying degrees of abstraction and approximation.

Parsing processes and strategies

Many of the most fundamental distinctions about parsing made in psycholinguistics can be interpreted as theories about the available syntactic processes. These include bottom-up and top-down parsing (Chomsky & Miller, 1963; Kimball, 1973), head-driven (Pritchett, 1988), left-corner (Aho & Ullman, 1972; Johnson-Laird, 1983), licensing (Abney, 1989), and other mixed strategies. For example, a strong commitment to head-driven parsing implies an architectural set of processes that includes projection from lexical heads and attachment to existing nodes, but excludes predictive processes that create new syntactic nodes whose lexical heads have not yet arrived.

Single-path vs. multi-path and serial vs. parallel

Single-path vs. multi-path parsing is another important distinction in parsing processes and representations. However, it should not be confused with parallel vs. serial processing. Whether processes occur in serial or in parallel is a *control* issue, not a distinction for processes per se. On the other hand, whether a parser pursues a single interpretation or multiple interpretations "in parallel" is a process and representation issue, and not a control issue.

The reason is that multi-path parsing requires a different set of processes and representations—it's not just single-path parsing run in parallel. At each local ambiguity or choice point, multi-path parsing requires that one of two things happens. Either a new copy (or copies) is made of the current structure, so that the two (or more) paths can proceed in parallel, or else the parser updates some sophisticated representation, like a chart (Earley, 1970), which permits existing structure to be used in multiple ways without copying. Both cases require additional processes and representations beyond that required by single path parsing.

Covering the range of comprehension functions

A complete architecture for sentence comprehension must support not only syntactic structuring, but also semantic interpretation, referential processing, and reanalysis as well. For example, it is a brute fact about natural language that any sentence processor must sometimes revise its interpretation of ambiguous material in the face of incoming disambiguating information. Lewis (1997a) distinguishes four ways of realizing reanalysis functions: (1) backtracking (overt and covert) to prior decision points; (2) selection (and disposal) from parallel alternatives; (3) monotonic refinement of abstract commitments; and (4) repair of existing structures. The choice of reanalysis processes depends on the processes assumed for initial structure building; for example, a single-path parser requires either a backtracking or repair solution. The important point here is not what choice is best[3], but that these choices must be made: reanalysis is a required function in language processing.

Chronometric properties

Specifying how long processes take is important for relating the architectural theory to the real time course of human sentence processing. Despite the widespread use of sophisticated chronometric techniques, most psycholinguistic theories say nothing quantitative about the time course of processing, seldom venturing beyond qualitative claims such as "revision takes time". The READER model of Thibadeau, Just and Carpenter (1982) was an exception, though the model was only used to parametrically fit existing reading time data, not generate new predictions

3.4 Specifying memories

Sentence comprehension requires a short-term working memory (STM) to hold the partial products of comprehension, and a longer-term memory (LTM) to hold lexical and grammatical knowledge or skill. We can characterize the architecture of memory in three important ways: (1) Identifying different kinds of memory systems and their unique coding schemes or memory units; (2) specifying the nature of the acquisition and retrieval processes; and (3) specifying how memory is limited in its capacity to carry out its required functions.

Kinds of memory

Perhaps the most fundamental architectural question about memory in sentence processing is: How many kinds are there, and what does a "kind" mean? Different memories may have different characteristic decay rates, coding schemes, refresh processes, and acquisition

[3] As argued in Lewis (1997a), there are good computational reasons to favor the repair solution.

and retrieval processes. We have already assumed at least a functional division between STM and LTM, but many other divisions have been proposed as well. For example, within working memory, there is evidence for independent phonological (Baddeley, 1990), visual (Logie et al., 1990), and semantic (Potter, 1993) short-term stores. There are also behavioral and neuropsychological double dissociations between phonological (classic verbal) working memory and syntactic working memory , providing strong evidence for some kind of architectural independence (Larkin & Burns, 1977; Lewis, 1996b; Martin, 1993; Potter, 1982). In a complex task such a language comprehension, all of these memories will interact to carry out the task.

Acquisition and retrieval mechanisms

What are the processes that acquire memories or change them? For STM, they are the syntactic, semantic, referential and reanalysis processes that manipulate structures in working memory. For long-term memory, the question is what are the learning mechanisms that acquire knowledge and skill for comprehension? Even if we grant that there are special aspects of first-language acquisition, vocabulary and comprehension skill acquisition go on beyond the initial acquisition years. In short, it makes sense to treat adult comprehension as a cognitive skill that may share at least some processing characteristics with other cognitive skills (Bever, 1970). Adopting such a view permits us to bring to bear general principles and mechanisms such as the power law of learning (Newell & Rosenbloom, 1981), encoding specificity (Tulving, 1983), or statistical rule tuning (Anderson, 1993; Mitchell et al., 1995).

Similarly, an architecture should specify how memory is indexed and retrieved or activated on-line. Many psycholinguistic theories incorporate principles of declarative memory retrieval, for example frequency-modulated activation (Anderson, 1993; MacDonald et al., 1994). But if we treat adult comprehension as a skill, we can ask whether principles of procedural knowledge retrieval apply as well, such as transfer by identical elements (Singley & Anderson, 1989), or encoding specificity.

Limitations of memory

Capacity limitations are perhaps the most familiar way that the architecture of memory, particularly short-term memory, shows through empirically. The role of STM was relatively neglected after the initial work spurred by Miller & Chomsky (1963), but some recent research has revived interest in memory limitations in comprehension, and the area continues to grow more active (Carpenter & Just, 1989; Just & Carpenter, 1992; Gibson, 1991; Lewis, 1996b). An important point to take from this work is that limitations in working memory may play a much wider role in explaining psycholinguistic phenomena than previously thought, affecting not only the parsing of deep embeddings, but also structural and lexical ambiguity resolution and recovery from garden paths. Limitations must show up in long-term memory as well, though there has been little work done to explore how such limitations could manifest themselves in the functions of sentence processing.

3.5 Specifying control structure

Specifying an architecture for comprehension requires specifying how the computational processes unfold over time—identifying the control structure. The control structure has implications for many important issues in sentence processing, including ambiguity resolution, automaticity, serial vs. parallel processing, and the relation of parsing and interpretation to central cognition. Indeed, one could argue that it is the control structure that gives an architecture its distinctive shape.

There are five key issues in specifying control structure: (1) specifying how processes are initiated; (2) specifying how processes communicate information to other processes; (3) specifying what parts of the control are fixed and what parts depend on variable content; (4) identifying multiple streams of control; and (5) specifying the relation of the control of comprehension processes to the control structure of central cognition.

Initiation of processes

An architecture's control structure specifies how processes are triggered or initiated. The most familiar way involves passing control from one step to the next in a procedural programming language, or passing control from a calling routine to a subroutine. This is the kind of control flow inherent in most standard programming languages. A modular pipeline architecture represents another kind of control structure: The processes become active as soon as their required inputs are present. This automaticity and independence of process initiation was one of the key distinguishing characteristics of modular systems proposed by Fodor (1983).

Communication of data between processes

An architecture's control structure specifies how data is communicated between processes. In a standard procedural subroutine language, data communication happens by packaging a fixed set of input variables to be passed from the calling routine to the subroutine. The communication is a narrow, directed pathway from one particular process to another. Each process or subroutine has access only to the set of variables passed to it by the calling routine[4]. A modular architecture is similar in some respects, because the communication pathways are fixed in advance. This hardwiring of communication is what gives rise to another principal characteristic of modular systems: information encapsulation. As Fodor (1983) went to lengths pointing out, information encapsulation should permit more efficient computation. The functional disadvantage is a set of processes that may ignore relevant knowledge sources.

Fixed vs. variable aspects of control

An architecture's control structure specifies what aspects of control are fixed, and which are variable. In a standard procedural programming language, the *program* represents variable

[4]Of course, this limited access can be violated by the use of global variables, but this is generally considered bad programming practice.

content which determines the control flow. A different program implies a different flow of processes. However, the program is also something that is fixed in advance of execution, so that the control decisions are not made at run-time. Conditional (IF-THEN) statements are one way that control flow can be made more flexible, but the alternative paths are nonetheless predetermined and fixed by the programmer.

Modular architectures often fix some control strategy to compensate for the loss of guiding information due to the restricted communication pathways. In single path parsing, control strategies such as minimal attachment (Frazier, 1987) correspond to specifying a fixed-in-advance scheme for ambiguity resolution[5]. More flexible control structures underly weakly interactive models such as Altmann and Steedman's (1988), which assume that variable content (e.g., referential context) can affect the control of parsing processes.

The rigidity of traditional control structures in computer science led Newell (1962, 1973) to consider production systems as an alternative control scheme. In a production system, the control flow is not fixed in advance by a program or script; instead, the set of active processes is determined on a moment-by-moment basis by the contents of the working memory.

Multiple streams of control (serial vs. parallel)

An architecture's control structure specifies what happens in parallel, and what happens serially. A standard programming language enforces a serial control structure; each process completes before the next is invoked. There is a single thread of control. By contrast, a modular pipeline architecture assumes a great deal of parallel processing, at least at the grain size of the modules. Each module works in parallel and independently of the others. The same holds for parallel productions systems. In both kinds of architectures, the only seriality emerges is that imposed by the inherent data dependencies in the task: if process (module or production) B requires some of process A's outputs, then B cannot be initiated

[5]Understanding precisely the set of architectural mechanisms implied by minimal attachment is no small undertaking. Without going into further details here, suffice it to say that all the possible architectural interpretations involve some kind of fixed control scheme.

until A produces the required outputs. The question for psycholinguistics is precisely what mix of serial (if any) and parallel processing underlies sentence comprehension.

Sentence processing and central cognition

A complete architecture for sentence processing must specify how the control structure for comprehension processes is related to the control structure of central cognition. The control of central processes is typically characterized as serial, not automatic, and open to any kind of variable content (Newell & Simon, 1972; Fodor, 1983), in sharp contrast to comprehension processes which are often characterized as automatic, parallel, depending on a fixed and limited information pathways, and with a control flow independent of central cognition—i.e., modular.

If one accepts this characterization of sentence processing control structure, at least to some degree, there are still a number of important questions remaining, some of which we raised earlier: How can these automatic processes be deployed under the deliberate control of central cognition, and at what grain size? The answer to these questions must be architectural, in particular, a specification of the connections between the control structure of sentence processing and the control structure of cognition.

3.6 Architectural constants

Simon (1974), Newell (1990), Miller (1956), and others have championed the search for a small set of constants that characterize human cognition. Examples include the time required to fixate a new item in long term memory (Newell & Simon, 1972), the size of a "chunk" (Simon, 1974), the decay rate of phonological STM (Baddeley, 1990), the time to make an elementary decision (Newell, 1990), and of course the celebrated size of STM (Miller, 1956).

Similarly, a theory of the architecture for comprehension should not only specify the qualitative shape of the architecture, but also specify the quantitative constants that characterize human sentence processing. The field of sentence processing (excluding the lexical access literature) has ielded few if any such architectural constants. Some candidates might include: What is the capacity of syntactic short-term memory? How long does a syntactic attachment take? How long does an elementary revision process take?

3.7 Summary: Some overlooked issues in architecture

Architectural theories must specify processes, memories, and control structure to be functionally complete. This view of architecture has led us to identify some issues that are often overlooked in sentence processing theories:

1. Many theories, in particular modular theories, are incomplete because they do not specify the nature of the memories used in parsing and interpretation.

2. Most theories do not specify explicit processes for reanalysis in the face of disambiguating information.

3. Most theories make little contact with general principles of human memory and skill (with statistical rule-tuning and frequency-based accounts being notable exceptions).

4. Most theories of the control of parsing and interpretation processes do not specify how automatic and controlled processes interact, or how automatic processes can be deployed under deliberate control (e.g., in service of deliberate reanalysis).

5. Most sentence processing theories do not identify the loci of individual differences in parsing and interpretation, whether based on architectural variation or knowledge differences. (Just & Carpenter (1992) and Perfetti (1985) are notable exceptions).

4 NL-Soar: An architectural theory based on Soar

This section describes an architecture for sentence processing that makes a set of specific claims about processes, memories, and control structure. The next section explores the empirical implications of this architecture. In many ways, the model focuses on traditional concerns in parsing: ambiguity resolution and garden path effects. But in other ways, it is an attempt to push in new directions to address the concerns of theoretical incompleteness raised in the last section.

The approach is to reduce the degrees of freedom in building the model by adopting as a starting point the assumptions of an existing theory of human cognitive architecture, Soar (Newell, 1990). Soar is not merely an implementation language for the computational model, but plays a fundamental theoretical role and accounts for many of the model's novel features and predictions. To the extent that we can derive principles of language processing from more general principles of cognition, we can increase the explanatory power of the theory.

4.1 Brief overview of Soar

What does it mean for NL-Soar to be based on Soar? It means that NL-Soar specifies how the functional requirements of comprehension are computationally realized within the architectural mechanisms of Soar. The NL-Soar theory takes these mechanisms as given—theoretical hypotheses independently motivated by other functional and empirical considerations.

The key features of Soar can be summarized as follows. All cognitive activity occurs in *problem spaces* (Newell, 1980b; Newell & Simon, 1972), where *operators* are applied to *states* to make progress toward a goal. The current problem space context is represented in a short-term working memory. All knowledge that guides behavior (e.g., search control) is held in a long term *recognition memory.* This memory consists of a set of *condition-action associations* (productions), which continually match in parallel against the working memory.

Soar's control structure is open and flexible. The flow of control is not fixed in advance, but is a function of whatever knowledge can be assembled at the time of the decision. The control structure is realized by an *elaborate-decide* cycle. In the *elaboration phase*, all associations whose condition patterns match the current working memory fire, retrieving knowledge about the current situation. These associations may propose new operators or problem spaces, apply operators (changing the state), or declare preferences about the relative desirability of operators and spaces. Associations fire automatically in parallel, or in sequence if there are inherent data dependencies. Next, a fixed *decision phase* interprets the retrieved preferences to determine the next step in the problem space—for example, deciding to apply one operator rather than another, or deciding to change problem spaces.

Sometimes the immediate knowledge retrieved from the recognition memory is incomplete or inconsistent in some way, and no more progress can be made. Then Soar has reached an *impasse*. For example, Soar might reach an impasse if three operators are proposed but no immediate knowledge is available to select which one to apply next. Soar responds to an impasse by automatically generating a subgoal to acquire the missing knowledge in other problem spaces. The behavior in the sub-problem spaces is guided in exactly the same manner as all behavior—by associations in long-term memory.

As knowledge is acquired from impasse-triggered problem solving, Soar's experiential learning mechanism, *chunking*, builds news associations in long-term memory which capture the results of the subgoal. The next time Soar encounters a similar situation, the new associations should allow it to recognize immediately what to do, avoiding the impasse. Soar is therefore an experience-based learner, continually making the shift from deliberation to recognition.

In terms of our previous account of architecture, Soar is clearly a complete functional architecture: it specifies processes (production firings to change WM, the learning mechanism to add to LTM), memories (the declarative working memory, the associational production memory), and control structure (the automatic, parallel production match, and the serial, controlled decision cycle).

4.2 Primitive and composed processes for parsing and interpretation

Table 1 summarizes the NL-Soar theory. Since all cognitive activity in Soar takes place by applying operators in problem spaces, comprehension in NL-Soar is accomplished by a set of *comprehension operators* that apply to the incoming linguistic input. The operators incrementally build two representations in working memory: a parse tree representing the syntactic structure of the input, and a situation model (Johnson-Laird, 1983; Kintsch et al.,

TABLE 1: Basic characteristics the NL-Soar theory. NL-Soar is the result of applying the Soar architecture to the task of efficiently comprehending language in real-time, with a minimal set of functional mechanisms.

1. Comprehension is realized by rapid cognitive operators (~50 ms) that incrementally build representations in working memory of syntax, meaning, and reference. *(Process)*

2. Comprehension is single path, with a cue-driven repair process that corrects (some) misinterpretations. *(Process)*

3. Working memory for syntax is a functionally minimal and efficient structure for discriminating syntactic nodes. *(Memory)*

4. Control of comprehension is a mix of controlled serial and automatic parallel processes. The serial control is potentially a function of any knowledge, and open to modulation by learning. *(Control)*

1990), representing the semantic and referential content of the utterance. There are three major classes of comprehension operators, corresponding to the major functions of parsing, semantic interpretation, and reference resolution.

A single comprehension operator may be composed of several primitive construction processes realized by individual association firings. For example, there might be a specific operator for creating an NP and attaching it as an object of a preposition. This operator would be realized by a ripple of association firings, each corresponding to some primitive process of projection of syntactic nodes from lexical heads, or linking two nodes via an X-bar (Chomsky, 1981) structural relation. The set of comprehension operators is not fixed in advance, nor is it specified explicitly by the NL-Soar theory. Rather, the theory posits that the set of operators is acquired as a function of linguistic experience. Other comprehension operators are composed in a similar way. For example, semantic interpretation operators are composed from a primitive set of situation model constructors (see (Lewis, 1993) for details).

4.3 Reanalysis by limited cue-driven repair

Soar adopts the *single-state assumption*: only one state per problem space is accessible in working memory at any given time. There is no architectural support for maintaining prior states in service of backtracking (Newell, 1990). Since each state in NL-Soar contains a representation corresponding to a particular interpretation, NL-Soar is a single path comprehender. The critical functional question that this gives rise to is: What happens when the incoming input is inconsistent with the structure chosen for some locally ambiguous material?

Because previous states are not readily available, the correct interpretation must be recovered by either deliberately recomprehending the input, or *repairing* the existing structures in working memory. NL-Soar posits that there exists such an on-line repair process which is constrained to ensure both limited match expense and limited problem space search. (Later we will also consider deliberate reprocessing of the input as a method of recovery.)

The minimal amount of new mechanism required to effect an on-line repair of syntactic structure is an operator that breaks an existing structural relation. We call this operator *snip* (Lewis, 1992; Lehman, Lewis, & Newell, 1991). In fact, snip does not complete the repair, it just destroys a bit of structure and then lets other *link* operators take over to finish the job. As an example of how snip works, consider what happens with the subject/object ambiguity in (7) below:

(7) Thad knows Shaq is tall.

Shaq is initially attached in the complement position of *knows.* When *is* arrives, it is projected to the sentential complement node CP[6]. Next, a link is proposed to attach the CP in the complement position of *knows.* This proposal is made because *knows* can take sentential complements as well as nominals.

Figure 3 shows what happens next. The result is a momentary superposition of two separate interpretations, with two nodes competing for the same structural position (COMP-V'

[6]CPs are projected in the absence of overt complementizers, following Pritchett (1992).

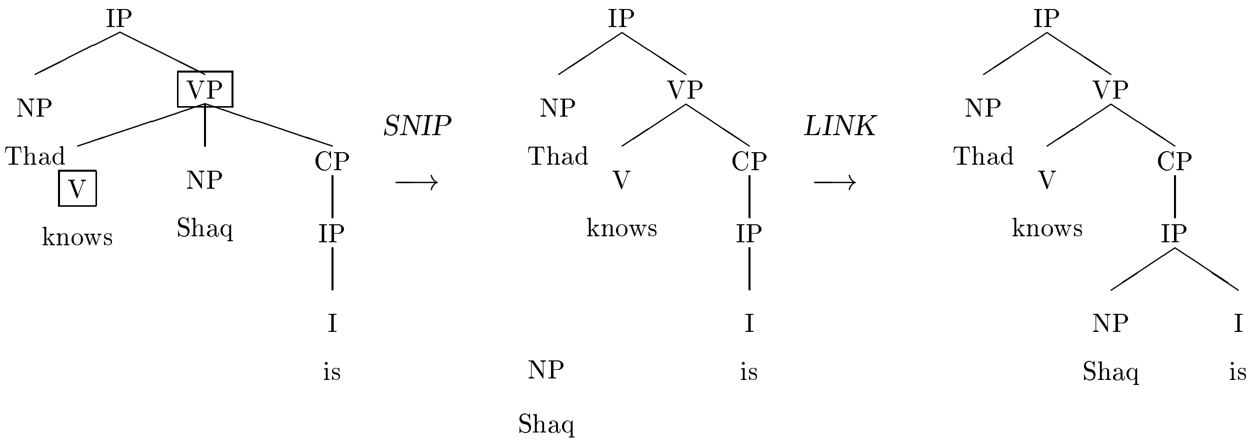

FIGURE 3: An example of simple cue-driven, destructive repair with the snip operator. The incoming CP competes for the same complement position as *Shaq*, triggering a snip to detach *Shaq*. Next, *Shaq* is attached as the subject of the incoming clause. The boxed nodes identify the locality of the inconsistency, which delimits the generation of snip operators.

of *knows*). This inconsistency triggers a local snip operator, which breaks the link between $[_{V'}$ knows] and $[_{NP}$ Shaq]. Next, $[_{NP}$ Shaq] is attached in subject position (SPEC-IP), and the repair is complete.

The initial attachment of the sentential complement was attempted despite the fact that another constituent was already occupying the complement position. This is the only way in which the attachment operators are nonstandard: they relax the constraint on uniqueness of complements and specifiers. Fodor & Inoue (1994) adopted a similar strategy, which they dubbed "attach anyway." The critical thing to note is that the constraints on the final output have *not* been relaxed. Only the set of processes and intermediate states that finally lead to the well-formed representation have changed.

The generation of snip is highly constrained. Snip is proposed only for structural relations that are *local* to a detected inconsistency (where local is defined precisely as the maximal projection). The boxed nodes in Figure 3 identify the locality of the inconsistency. This is a *cue-driven* theory of repair, because the mechanism relies on simple, local structural cues that something has gone wrong with the parse. There are good computational reasons for such a tightly constrained repair. A free generation of snip for every link in the parse tree has two undesirable consequences. First, it leads directly to a potentially large, undiscriminated set of operators in working memory. Such sets are a source of exponential match cost in the recognition memory (Tambe et al., 1990). Second, even for moderately-sized syntactic structures, the introduction of freely generated snips increases the problem search space significantly. We can summarize the snip theory as follows:

> **Automatic reanalysis is realized by a limited, cue-driven repair process. Repair works by locally and destructively modifying the given syntactic structure in working memory when an inconsistency arises, and then reassembling the pieces with the normal constructive parsing operators.**

§5.1 explores the implications of limited repair for strong and weak garden path effects.

4.4 Interference in short-term memory

NL-Soar's working memory is shaped by two concerns: meeting the minimal functional requirements of parsing natural language, and keeping the processing efficient. These concerns lead naturally to a simple interference theory of short-term memory.

Partial constituents are indexed in working memory by the structural relations that they may enter into with other constituents. The relations correspond to X-bar structural positions (complement of preposition, specifier of NP, etc; though another ontology of relations could be adopted.) Consider the prepositional phrase *with the dog*. We say that $[_{P'}$ with] *assigns* the COMP-P' (complement of preposition) relation, and $[_{NP}$ the dog] *receives* or fills the COMP-P' relation.

The representation in working memory is called the *H/D set*, for *Heads* and *Dependents*. The Heads part of the H/D set indexes nodes by the structural relations they may assign. The Dependents part of the H/D set indexes nodes by the structural relations they may receive. The H/D set (8) below corresponds to the situation in parsing *with the dog* after the $[_{NP}$ the dog$]$ has been formed, but before creating the complete PP:

(8)

Heads	COMP-P':	$[_{P'}$ with$]$
Dependents	ADJOIN-N':	$[_{PP}$ with$]$
	COMP-P':	$[_{NP}$ the dog$]$
	COMP-V':	$[_{NP}$ the dog$]$

Parsing is a bottom-up process of matching potential heads with potential dependents. For example, in (8) above, $[_{P'}$ with$]$ and $[_{NP}$ the dog$]$ may be joined by the structural relation COMP-P', since $[_{NP}$ the dog$]$ is indexed under COMP-P' in the Dependents set, and $[_{P'}$ with$]$ is indexed under COMP-P' in the Heads set.

How should the representation be limited to ensure efficient processing? As mentioned earlier, independent theoretical and empirical work on the computational complexity of the recognition match (Tambe, Newell, & Rosenbloom, 1990) has identified open, undiscriminated sets in working memory as the most significant source of match expense. An undiscriminated set is simply a set of elements in working memory indexed by a single relation or attribute. With such open sets, the recognition match becomes exponentially expensive and therefore psychologically and computationally implausible as the basis for efficient memory retrieval.

Large, undiscriminated sets may be created when multiple constituents are indexed by a single syntactic relation. For example, consider the right-branching structure in (9):

(9) Amparo thinks that Seth believes that John knows ...

Such right-branching can lead to a unbounded set of nodes indexed by a single relation (in this case the complement of verb relation on the Heads set):

(10)

Heads	COMP-V':	$[_{V'}$ thinks$]$, $[_{V'}$ believes$]$, $[_{V'}$ knows$]$ ...

We can eliminate such open sets by limiting each relation to a small fixed number of nodes. But how many?

To be able to at least parse basic sentential complements such as *I think that John likes Mary*, two nodes per relation are required. This is the minimum capacity required to realize the fundamental capability of natural language to compose new propositions from existing ones (Lewis, 1997b)[7].

The NL-Soar theory makes the strong assumption that this minimal functionality characterizes the capacity of human syntactic short-term memory:

> **Each syntactic relation in working memory indexes at most two nodes, the minimum required to achieve propositional composition. This limitation yields a similarity-based interference theory of syntactic STM.**

Later, we will see how precisely how the interference effects arise as a function of the structural similarity of working memory contents.

4.5 Control: automatic but flexible

What is the control structure of comprehension? As we saw in the earlier architectural analysis, this question is at the heart of the debate on modularity. NL-Soar provides a well-defined answer: the control structure of comprehension is the flexible control structure of Soar. This has several immediate implications. There is a mix of parallel and serial processes. The parallel firing of productions is automatic, and the control of the serial operators is open to modulation by multiple knowledge sources. There are no fixed architectural barriers to the kind of knowledge that may be brought to bear in selecting what path to pursue. Furthermore, the control knowledge is open to continuous modification by chunking. Any linguistic decision point can potentially be modified if new associations are learned and brought to bear at the appropriate time.

Because chunking is a general learning mechanism that operates over multiple impasse types, each aspect of comprehension is open to learning and improvement. New comprehension operators are learned by acquiring new associations that generate (propose) and apply

[7]This can be seen in (9) above, because *thinks* and *believes* must both be available momentarily on the COMP-V' heads relation: when *thinks* is attached to *believes*, it must be available to assign its complement, and *believes* must be accessible in order to semantically interpret its complement. Other simple propositional compositions such as relative clauses also require two nodes per relation.

(implement) the operators. The model can also learn new search control associations to guide parsing or semantic interpretation. The process of reference resolution results in associations that constitute a recognition memory of the content of a discourse (Lewis, 1993). Together, this growing collection of proposal, application, and control associations for syntactic, semantic, and referential operators constitutes the automatized comprehension skill. To summarize the control theory:

> **Control of comprehension is two layered, with a mix of serial and parallel processes. The parallel firing of primitive associations is automatic and uncontrolled; the serial selection of composed operators is controlled and a function of potentially any knowledge source. The control of operators is open to modulation by learning.**

5 Testing the theory empirically

The theory has now been described in sufficient detail that we can explore a few of its empirical implications. Each of the following sections show how some aspect of the architecture (e.g, the repair process, or control structure) makes predictions concerning some psycholinguistic phenomenon (e.g., strong garden path effects), in particular, those phenomena that underly the set of behavioral contrasts set up in Figure 1. There is space here to present only a few examples of the application of the theory; for fuller treatments see (Lewis, 1993; Lewis, 1996b; Lewis & Lehman, 1997).

5.1 Limited repair: Easy vs. difficult garden path effects

We claimed earlier that a theory of human parsing must answer two related questions: (1) How do people interpret locally ambiguous material, and (2) How do people revise their interpretations of ambiguous material based on later disambiguating information? Limited repair is an answer to the second question, and explains how the human sentence processor can sometimes rapidly leap off the garden path and find its way back onto the path to a successful parse. This section explores predictions that the snip-based repair

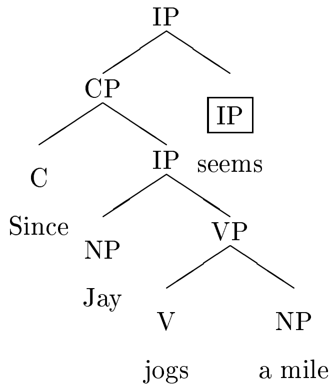

FIGURE 4: Failure to repair a subject/object ambiguity.

mechanism makes across a range of structural ambiguities, including some that derive from lexical categorial ambiguities. The structure presented here are a subset of the roughly 70 structures analyzed in (Lewis, 1993; Lewis & Lehman, 1997).

Subject/object ambiguities

We have already seen how the theory handles an easy subject/object ambiguity (7). As an example of how the theory predicts a difficult garden path, consider the following from (Frazier & Rayner, 1982):

(11) Since Jay always jogs a mile seems like a short distance to him.

Here, *a mile* is taken initially as the complement of *jogs*, just as in (7). Because *jogs* does not take sentential complements, the initial phrase *Since Jay jogs* is adjoined to the incoming *seems.* In fact, this is its correct final position. However, [$_{IP}$ seems] is still missing its subject. But in this case a snip operator is not generated for the complement relation between [$_{V'}$ jogs] and [$_{NP}$ a mile], because the relation is not local to the detected inconsistency (the missing obligatory subject). This situation is shown in Figure 4, with the boxed nodes again representing the locality of the inconsistency. As a result, [$_{NP}$ a mile] is not reanalyzed as the subject of [$_{IP}$ seems], and the necessary repair fails.

Difficult subject/object garden paths such as (11) above show up cross-linguistically as well; for example, in Mandarin (Gorrell, 1991) and Hebrew (Pritchett, 1992). The explanation is the same as in the English example: the local snip is not generated to remove the object NP from complement position.

Main verb/relative ambiguities

We now explore variations on the classic relative clause garden path. Consider the canonical example (Bever, 1970):

(12) The horse raced past the barn fell.

In the structure for the main verb interpretation of (12), the inflectional features that head the IP phrase adjoin to the verb, leaving a trace in head of IP. (This joining of inflectional features to the verb is assumed in some form by many syntactic theories; e.g., McCawley (1988a) calls it *Tense-hopping*, and assumes an adjunction structure like the one presented here.) On the other hand, passive forms like *driven* and *raced* are untensed. In the reduced relative reading of *the horse raced past the barn*, which uses the passive interpretation of *raced*, the inflection is not present.

Consider now the repair required to successfully parse (12). The main verb structure must be repaired into the reduced relative structure. This involves snipping the adjoined inflectional features. When *fell* arrives and is projected to VP, the only place it may attach is in complement position of I'. This produces an inconsistency local to the IP, as shown in Figure 5. However, this fails to trigger all the required snips; in particular, the crucial inflection adjunction is left undisturbed, so the passive reading cannot be recovered.

The intervening modifier [$_{PP}$ past the barn] is irrelevant to this explanation. Thus, the theory correctly predicts the existence of very short reduced relative garden paths (Kurtzman, 1985; Abney, 1989):

(13) The boat floated sank.

Such examples are one demonstration of the independence of garden path effects from length effects (Pritchett, 1992; Lewis, 1993).

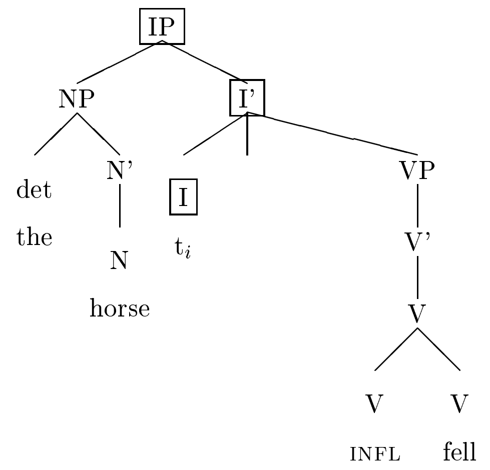

FIGURE 5: The main verb/reduced relative garden path effect. When *fell* arrives and is projected to VP, the only place it may attach is in complement position of I'. This produces an inconsistency local to the IP (denoted by the boxed nodes). However, this fails to trigger all the required snips, in particular, the crucial inflection adjunction is left undisturbed, so the required passive reading cannot be recovered.

Surprisingly, not all main verb/relative clause ambiguities produce garden path effects. Mazuka et al. (1989) present an interesting unproblematic Japanese construction involving a main verb/relative ambiguity:

(14) a. Roozin ga kodomo o yonda
old man NOM child ACC called
(The old man called the child.)

b. Roozin ga kodomo o yonda zyosee to hanasi o sita.
old man NOM child ACC called woman with talk ACC did.
(The old man talked with the woman who called the child.)

In (14), the NP NP V sequence *roozin ga kodomo o yonda* is interpreted as the main clause *The old man called the child.* In (14), the relative clause reading is required, disambiguated by the appearance of [$_{NP}$ zyosee]. Unlike the familiar English main verb/reduced relative ambiguity, NL-Soar can repair the structure in (14). The main clause interpretation is pursued initially, with $[_{NP}$ roozin$]$ in subject (SPEC-IP) position and $[_{NP}$ kodomo o$]$ in complement position of $[_{VP}$ yonda$]$. Next, $[_{NP}$ zyosee$]$ arrives and the CP adjoins to $[_{N'}$ zyosee$]$ as a modifying clause (unlike the English version, the relative clause is active, not passive, and therefore the clause remains tensed). The appropriate traces are generated in SPEC-CP and SPEC-IP position, in the same manner as English relative clauses. The SPEC-IP trace creates a local inconsistency at the IP node, triggering a snip of $[_{NP}$ roozin$]$. $[_{NP}$ roozin$]$ is now available to attach as the subject of the incoming $[_{IP}$ to hanasi o sita$]$, and the repair succeeds.

Lexical categorial ambiguities

Lexical ambiguity often gives rise to structural ambiguity. Consider the basic noun/adjective ambiguity in (15):

(15) a. The square is red.

b. The square table is red.

When *square* arrives, both categories are retrieved in parallel from the lexicon, and NP and AP nodes are projected. Next, the determiner *the* is attached in SPEC-NP position, forming $[_{NP}$ the square$]$. Then *table* arrives and is projected to NP. Next, the adjective phrase $[_{AP}$ square$]$ is adjoined to $[_{N'}$ table$]$. Each syntactic link is well-formed, but two mutually incompatible bits of structure have been produced, since the single token *square* cannot simultaneously be an adjective and a noun.

A snip operator is immediately triggered by the inconsistency of syntactic structure attached to competing senses of the same lexical token. A locally generated snip breaks the SPEC-NP link between the determiner and $[_{NP}$ square$]$. Next, a link operator attaches the determiner in specifier position of $[_{NP}$ table$]$, forming $[_{NP}$ the square table$]$. This complete the repair. In this way, the snip operator extends naturally to handle repairs involving lexical ambiguity.

We have now seen all three kinds of structural cues that trigger the snip operator:

1. Multiple phrases competing for the same structural position.
2. Missing obligatory phrases (e.g., missing subjects)
3. Attachments to competing categorial interpretations of the same lexical item.

NL-Soar also handles *multiple* lexical ambiguities. For example, noun/verb ambiguities may be preceded by adjective/noun ambiguities without causing difficulty (Milne, 1982; Pritchett, 1992):

(16) a. The square blocks the triangle.

b. The square blocks are red.

NL-Soar effects the repairs in (16) in the same manner as (15); see Lewis (1993) for details. Yet some noun/verb ambiguities do cause difficulty. If the unproblematic ambiguity in (16) is followed by a reduced relative, the result is a strong garden path (Milne, 1982; Pritchett, 1992):

(17) The building blocks the sun faded are red.

Suppose that *blocks* is initially taken as the main verb, and *sun* as the complement. When *faded* arrives, it can be attached as a reduced relative modifying *sun*. Once *are* is projected to an IP, no additional attachments are possible. Furthermore, there are no local inconsistencies to generate a snip, so the repair is never initiated.

Summary

As a theory of reanalysis, snip-based limited repair has a number of important features:

1. The theory is formulated *independently of what guides the initial choice* at the ambiguity point. Thus, the theory classifies structure types as *potential* garden paths, not definite garden paths. The repair theory determines whether or not a given inconsistent structure can be repaired. Whether a local ambiguity gives rise to a strong garden path effect in any particular context is a function *both* of the ambiguity resolution process itself, and the efficacy of the repair.

2. The theory is *cue-driven* theory of repair, because it depends on a few simple local cues to trigger the repair process.

3. The theory is a *functional* theory, in that it posits mechanisms to efficiently carry out the functions of reanalysis. NL-Soar is not a linguistic metric that distinguishes easy and difficult ambiguities. It is an implemented computational theory that classifies certain sentences as potential strong garden paths because it may fail to parse those sentences.

4. The theory posits that *on-line repair is not costly*; in fact, the prediction is that these small local repairs are a frequently occurring part of normal sentence comprehension.

5. The theory posits a *minimal amount of additional new mechanisms and control* to carry out the repair. The only new operator is snip; repair is carried out by snip and a sequence of the existing parsing operators. The only important new piece of control knowledge is a rule that says roughly: *don't relink what you just snipped.*

5.2 Interference in STM: Easy vs. difficult embeddings

This section explores some interesting predictions that the limited STM makes in the area of cross-linguistic embeddings. The goal is to provide just a glimpse of the empirical richness of the domain; for a comprehensive empirical exploration, see the 50+ constructions analyzed in Lewis (1993) and Lewis (1996b).

Consider again the classic double-embedded relative clause (Miller & Chomsky, 1963):

(18) The boy that the man that the woman hired hated cried.

The Dependents set must index the three initial NPs under spec-IP, since all three NPs will eventually occupy subject position:

(19)

Dependents	spec-IP:	$[_{NP}$ the boy],$[_{NP}$ the man],$[_{NP}$ the woman]

This exceeds the posited capacity of two nodes per relation (§4.4). Failure will occur at one of the final verbs (which verb depends on which NP is dropped from the H/D set). It is important to note that failure does not occur just because nodes are removed from the set. Just as in standard short-term memory tasks, interference is only a problem if the item that is interfered with must be accessed later.

The uniform limit of two nodes per relation in the H/D set theory does not entail a prohibition against all double center embedding. If that was the case, it would essentially be equivalent to Kimball's (1973) Principle of Two Sentences, and fall prey to the same empirical problems (Gibson, 1991). Consider the pseudo-cleft construction, which has something like a Wh-clause in subject position:

(20) a. What the man saw was a dog.

b. $[_{IP}$ $[_{NP}$ $[_{CP}$ What the man saw]] was a dog]

The initial Wh-clause in pseudo-clefts is an indirect question (Baker, 1989; McCawley, 1988b). The interaction of this structure with the H/D set leads to some interesting predictions. Because the initial Wh-word does not occupy spec-IP, it should be possible to embed an additional relative clause within the indirect question without causing difficulty. This prediction turns out to be true (Gibson, 1991):

(21) a. What the woman that John married likes is smoked salmon.

$[_S$ What $[_S$ the woman that $[_S$ John married] likes] is smoked salmon.]

b.

Dependents	spec-IP:	$[_{NP}$ woman], $[_{NP}$ John]
	spec-CP:	$[_{NP}$ what]

A final example in Japanese will illustrate the theory's application to a typologically distinct language. Though the English (18) causes difficulty with three stacked NPs, Japanese sentences that stack up to five initial NPs are acceptable to native speakers:

(22) John-wa Bill-ni Mary ga Sue-ni Bob-o syookai sita to it-ta.

John TOP Bill DAT Mary NOM Sue DAT Bob ACC introduced COMP say PERF.

(John said to Bill that Mary introduced Bob to Sue.)

Although such sentences are surely complex, they do not cause the failure associated with (18). The H/D set can handle structures such as (22) because no single structural relation must buffer more than two NPs:

(23)	Dependents	spec-IP:	$[_{NP}$ John], $[_{NP}$ Mary]
		comp-V':	$[_{NP}$ Bob]
		comp2-V':	$[_{NP}$ Bill], $[_{NP}$ Sue]

Summary

Apart from the cross-linguistic empirical coverage, the theory of syntactic STM presented here has a number of important features:

1. The theory is a *functional* architectural theory of memory; it is not a linguistic metric that distinguishes easy and difficult embeddings (e.g., Gibson 1991). It classifies certain sentences as difficult because the posited structure of memory fails to support a successful parse of those sentences.

2. The theory embodies a general principle of short-term memory limitation, *similarity-based interference*, which has been demonstrated in a number of different tasks and modalities, ranging from visual stimuli to sign language (Baddeley, 1966; Conrad, 1963; Magnussen et al., 1991; Wickelgren, 1965; Shulman, 1970; Waugh & Norman, 1965; Poizner et al., 1981).

3. It correctly predicts that difficulty can arise independently of ambiguity (Blumenthal, 1966; Blauberg & Braine, 1974; Foss & Cairns, 1970; Hamilton & Deese, 1971; Larkin & Burns, 1977; Marks, 1968; Miller & Isard, 1964).

4. It correctly predicts that deep embeddings alone (even center-embedding) is insufficient to cause difficulty (see, e.g,. (9) and (21)).

NL-Soar's model of working memory captures two important theoretical convergences in recent accounts of syntactic short-term memory. First, the source of memory load is open or unsatisfied syntactic relations (Abney & Johnson, 1991; Gibson, 1991; Stabler, 1994a).

This leads naturally to a focus on stacking, rather than embedding per se, as the source of difficulty (Gibson, 1991; Hakuta, 1981; Mazuka et al., 1989). Second, increasing similarity makes things more difficult. This shows up clearly in the "broad but shallow" memory models of Reich (1969) and Stabler (1994a), and the self-embedding metric of Miller & Chomsky (1963).

5.3 Some implications of control structure

Recall from section 2 that all processing in Soar and NL-Soar consists of a sequence of decision cycles proposing, selecting, and applying cognitive operators. Local ambiguity manifests itself by the simultaneous proposal of a set of operators corresponding to the different interpretations at the ambiguous point. For example, in (1) above, at the verb *examined*, two operators are proposed: one corresponding to the main verb structure, and one corresponding to the relative clause structure. Ambiguity resolution then takes place in the same way that all operator selection is realized in Soar: by drawing on available search control productions. In the case of the ambiguity in (1), a search control production can test the proposed operators and the semantic content of the subject (e.g, *evidence* or *car*), and prefer the relative-clause operator (or disprefer the main-verb operator) in the appropriate contexts. Thus, NL-Soar can model the rapid, on-line effects of semantic or referential context observed in some studies.

However, nothing guarantees that such search control productions will be available. If the knowledge is present only through deliberate processing, there may not be enough time to perform all the inferences necessary to make the right selection. Under press of time, there may be no alternative but to select one interpretation by some default preference. In such a case, NL-Soar is behaving in a modular fashion since the required knowledge sources are not applied on-line.

These and similar kinds of limitations that arise in NL-Soar are a kind of forgetting due to retrieval failure. In other words, the required knowledge may be present, but because the correct set of cues are not assembled in working memory, the knowledge is not evoked. In this way, the limitations can be seen as emerging from the fact that Soar embodies

Tulving's (1983) encoding specificity principle: The cues required for retrieval are a subset of specific aspects of the environment at learning time.

(Lewis, 1993) describes in detail further implications of NL-Soar's control structure, including the ability to deliberately re-comprehend problematic ambiguities. This kind of flexible interplay between automatic and deliberate processing is a hallmark of Soar's control structure, but the fact that this structure nevertheless yields processing limitations such as the masking effect, and that these limitations have yield modular-like properties, is somewhat surprising giving Soar's origins in purely functional concerns.

6 Conclusions

This paper began by developing a set of empirical and functional constraints on information processing theories of sentence processing. The empirical constraints took the form of a set of behavioral oppositions that outline the impressive capabilities and severe limitations of human comprehension. The primary functional constraint was that sentence processing theories should take the form of complete computational architectures which specify the processes, memories, and control structures underlying human parsing and interpretation. This functional, architectural approach revealed several overlooked issues in sentence processing architecture.

The empirical and explanatory power of the approach was demonstrated by presenting NL-Soar, a computational architecture for sentence comprehension. The theory covers a broad range of sentence processing phenomena: strong garden path effects and easy ambiguities, difficult and acceptable embeddings, modular and interactive ambiguity resolution effects, and aspects of the time course of comprehension. In the areas of garden path effects and difficult/easy embeddings, the accounts are deep: the theory makes successful predictions on a large collection (> 100) of cross-linguistic structures (Lewis, 1993).

The model has a number of important features that help push sentence processing theory in new directions (cf. §3.7). Among the most prominent of these is reducing theoretical degrees of freedom by adopting the assumptions of an independently motivated theory of cognitive architecture. One of the yields of of such an approach was an explanation of how certain limitations in sentence processing embody some general principles of human memory and skill, in particular, similarity-based interference in STM, encoding specificity (Tulving, 1983), and *Einstellung* (Luchins, 1942). The model also simultaneously accounts for on-line effects of context and semantics, and subtle effects of syntactic structure that distinguish easy and difficult garden paths and embeddings. All of these implications and predictions flow from the interactions of some fairly simple core architectural assumptions: the two-layered parallel/serial control structure, the limited interference-based working memory, and the limited repair process.

Two objections might be raised to the theoretical approach advocated here. One might be called the *problem of irrelevant specification*, a problem with building computational models in general. The objection goes as follows. Computational models, because they force functional completeness, also inevitably lead to specification of irrelevant detail without empirical support. There are two responses to this. One response is to simply be careful about abstracting out the essential theoretical claims and identifying their empirical support. That has been the intention here; the underlying computational simulation undoubtedly contains many details that are not relevant to the core theoretical claims. The second response is inherent in the architectural approach itself: an architectural theory makes principled distinctions between the fixed structure that carries the theory, the variable content that is posited for some particular task, and the irrelevant implementation technology. Thus, the theoretical status of an architectural simulation is much clearer than is the case for information processing simulations in general (Newell, 1990; Pylyshyn, 1984).

The second objection might be called *the problem of anything-goes cognizers.* Many theorists have warned against abandoning highly constrained modular theories in favor of anything that approaches more general problem solving machinery (Fodor, 1983; Fodor, 1990; Forster, 1979). Under this view, building a sentence processing architecture within a general cognitive engine like Soar is exactly the wrong thing to do because it opens the door to unconstrained theorizing. But the general injunction should be against theories with too many degrees of freedom, not against general architectures per se. In fact, the approach taken here, to adopt Soar as a starting point, is precisely a way of eliminating theoretical degrees of freedom: the control structures and memories are independently motivated givens. Furthermore, this approach also seeks to find contact with other principles of cognitive processing (e.g., the memory principles outlined above), and thereby increase explanatory power. In the specific case of NL-Soar, the underlying general architecture (Soar) was the vehicle for making most of those connections, and actually led to a new theory that fits well with both the theoretical and empirical concerns of modularity (Lewis, 1996a).

In short, this kind of architectural approach begins to address the concerns of both Bever (1970), who wanted psycholinguistic theory to be grounded in general cognitive principles, and Forster (1979), who wanted psycholinguistic theory to be tightly constrained yet integrated with central processes. The general analysis and theory presented here are clearly just a first step down this path, but the initial results seem promising.

References

Abney, S. P. (1989). A computational model of human parsing. *Journal of Psycholinguistic Research*, 18(1):129–144.

Abney, S. P. & Johnson, M. (1991). Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20(3):233–250.

Aho, A. & Ullman, J. (1972). *The Theory of Parsing, Translation, and Compiling. Volume 1: Parsing*, volume 1. Prentice Hall, Englewood Cliffs, NJ.

Altmann, G. & Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, 30:191–238.

Anderson, J. R. (1983). *The Architecture of Cognition.* Harvard University Press, Cambridge, MA.

Anderson, J. R. (1993). *Rules of the Mind.* Lawrence Erlbaum, Hillsdale, NJ.

Baddeley, A. D. (1966). Short-term memory for word sequences as a function of acoustic, semantic, and formal similarity. *Quarterly Journal of Experimental Psychology*, 18:362–365.

Baddeley, A. D. (1990). *Human Memory: Theory and Practice.* Allyn and Bacon, Boston.

Baker, C. L. (1989). *English Syntax.* MIT Press, Cambridge, MA.

Bell, C. G. & Newell, A. (1971). *Computer Structures: Readings and Examples.* McGraw Hill, New York.

Bever, T. G. (1970). The cognitive basis for linguistic structures. In Hayes, J. R., editor, *Cognition and the Development of Language*, pages 279–362. Wiley, New York.

Blauberg, M. S. & Braine, M. D. S. (1974). Short-term memory limitations on decoding self-embedded sentences. *Journal of Experimental Psychology*, 102(4):745–748.

Blumenthal, A. L. (1966). Observations with self-embedded sentences. *Psychonomic Science*, 6:453–454.

Carpenter, P. A. & Just, M. A. (1989). The role of working memory in language comprehension. In Klahr, D. & Kotovsky, K., editors, *Complex Information Processing: The Impact of Herbert A. Simon*. Erlbaum, Hillsdale, NJ.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Chomsky, N. (1980). Rules and representations. *The Behavioral and Brain Sciences*, 3:1–15.

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

Chomsky, N. & Miller, G. A. (1963). Introduction to the formal analysis of natural languages. In Luce, D. R., Bush, R. R., & Galanter, E., editors, *Handbook of Mathematical Psychology*, volume II. John Wiley, New York.

Clark, H. & Clark, E. (1977). *The Psychology of Language: An introduction to psycholinguistics*. Harcourt Brace Jovanovich, New York.

Conrad, R. (1963). Acoustic confusions and memory span for words. *Nature*, 197(4871):1029–1030.

Cowper, E. A. (1976). *Constraints on Sentence Complexity: A Model for Syntactic Processing*. PhD thesis, Brown University.

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the ACM*, 13:94–102.

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

Ferreira, F. & Henderson, J. M. (1990). Use of verb information in syntactic parsing: Evidence from eye movements and word-by-word self-paced reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16:555–568.

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

Fodor, J. D. (1990). Thematic roles and modularity: Comments on the chapters by frazier and tanenhaus et al. In Altmann, G. T. M., editor, *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives.* The MIT Press, Cambridge, MA.

Fodor, J. D., Fodor, J. A., & Garrett, M. F. (1975). The psychological unreality of semantic representations. *Linguistic Inquiry*, 6(4):515–531.

Fodor, J. D. & Inoue, A. (1994). The diagnosis and cure of garden paths. *Journal of Psycholinguistic Research*, 23(5):407–434.

Forster, K. (1979). Levels of processing and the structure of the language processor. In Cooper, W. E. & Walker, E. C., editors, *Sentence Processing: Psycholinguistic Studies Presented to Merrill Garrett.* Lawrence Erlbaum, Hillsdale, NJ.

Foss, D. J. & Cairns, H. S. (1970). Some effects of memory limitation upon sentence comprehension and recall. *Journal of Verbal Learning and Verbal Behavior*, 9:541–547.

Frazier, L. (1978). *On comprehending sentences: Syntactic parsing strategies.* PhD thesis, University of Connecticut.

Frazier, L. (1987). Sentence processing: a tutorial review. In Coltheart, M., editor, *Attention and Performance XII: The Psychology of Reading.* Lawrence Erlbaum Associates, Ltd., East Sussex, U.K.

Frazier, L. (1990). Parsing modifiers: Special purpose routines in the human sentence processing mechanism? In Balota, D. A., Flores d'Arcais, G. B., & Rayner, K., editors, *Comprehension processes in reading.* Lawrence Erlbaum, Hillsdale, NJ.

Frazier, L. & Clifton, Jr., C. (1996). *Construal.* The MIT Press, Cambridge, MA.

Frazier, L. & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14:178–210.

Gibson, E. A. (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.

Gorrell, P. (1987). *Studies of Human Syntactic Processing: Ranked-Parallel Versus Serial Models.* PhD thesis, The University of Connecticut.

Gorrell, P. (1991). Subcategorization and sentence processing. In Berwick, R. C., Abney, S. P., & Tenny, C., editors, *Principle-based Parsing: Computation and Psycholinguistics.* Kluwer Academic, Dordrecht, The Netherlands.

Hakuta, K. (1981). Grammatical description *versus* configurational arrangement in language acquisition: The case of relative clauses in Japanese. *Cognition*, 9:197–236.

Hamilton, H. W. & Deese, J. (1971). Comprehensibility and subject-verb relations in complex sentences. *Journal of Verbal Learning and Verbal Behavior*, 10:163–170.

Johnson-Laird, P. N. (1983). *Mental Models.* Harvard, Cambridge, MA.

Juliano, C. & Tanenhaus, M. K. (1994). A constraint-based lexicalist account of the subject/object attachment preference. *Journal of Psycholinguistic Research*, 23(6):459–471.

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

Just, M. A. & 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.

Kintsch, W., Welsch, D., Schmalhofer, F., & Zimny, S. (1990). Sentence memory: a theoretical analysis. *Journal of Memory and Language*, 29:133–159.

Kurtzman, H. S. (1985). *Studies in Syntactic Ambiguity Resolution.* PhD thesis, Massachusetts Institute of Technology.

Laberge, D. & Samuels, S. J. (1974). Toward a theory of automatic information processing in reading. *Cognitive Psychology*, 6:293–323.

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

Lehman, J. F., Laird, J. E., & Rosenbloom, P. S. (1996). A gentle introduction to soar, an architecture for human cognition. In *Invitation to Cognitive Science*, volume 4. MIT Press.

Lehman, J. F., Lewis, R. L., & Newell, A. (1991). Integrating knowledge sources in language comprehension. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 461–466. Also in P. S. Rosenbloom, J. E. Laird, and A. Newell, eds., *The Soar Papers: Research on Integrated Intelligence*, MIT Press, Cambridge, MA, 1993.

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). *An Architecturally-based theory of Human Sentence Comprehension.* PhD thesis, Carnegie Mellon University. Available as technical report CMU-CS-93-226 from reports@cs.cmu.edu.

Lewis, R. L. (1996a). Architecture Matters: What Soar has to say about modularity. In Steier, D. & Mitchell, T., editors, *Mind Matters: Contributions to Cognitive and Computer Science in Honor of Allen Newell.* Erlbaum, Hillsdale, NJ.

Lewis, R. L. (1996b). Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research*, 25(1):93–115.

Lewis, R. L. (1997a). Leaping off the garden path: Reanalysis and limited repair parsing. In Fodor, J. & Ferreira, F., editors, *Reanalysis in Sentence Processing*. Kluwer Academic, Boston. In press.

Lewis, R. L. (1997b). A theory of grammatical but unacceptable embeddings. In revision.

Lewis, R. L. & Lehman, J. F. (1997). A theory of the computational architecture of sentence comprehension. Submitted manuscript, Ohio State University.

Logie, R. H., Zucco, G. M., & Baddeley, A. D. (1990). Interference with visual short-term memory. *Acta Psychologica*, 75:55–74.

Luchins, A. S. (1942). Mechanization in problem solving. *Psychological Monographs*, 54(6).

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994). The lexical nature of syntactic ambiguity resolution. *Psychological Review*, 101:676–703.

MacKay, D. G. (1966). To end ambiguous sentences. *Perception & Psychophysics*, 1:426–436.

Magnussen, S., Greenlee, M. W., Asplund, R., & Dyrnes, S. (1991). Stimulus-specific mechanisms of visual short-term memory. *Vision Research*, 31(7/8):1213–1219.

Marks, L. E. (1968). Scaling of grammaticalness of self-embedded English sentences. *Journal of Verbal Learning and Verbal Behavior*, 7:965–967.

Martin, R. C. (1993). Short-term memory and sentence processing: Evidence from neuropsychology. *Memory & Cognition*, 21(2):176–183.

Mazuka, R., Itoh, K., Kiritani, S., Niwa, S., Ikejiru, K., & Naitoh, K. (1989). Processing of Japanese garden path, center-embedded, and multiply left-embedded sentences. In *Annual Bulletin of the Research Institute of Logopedics and Phoniatrics*, volume 23, pages 187–212, University of Tokyo.

McCawley, J. D. (1988a). *The Syntactic Phenomena of English, Volume 1.* The University of Chicago Press, Chicago.

McCawley, J. D. (1988b). *The Syntactic Phenomena of English, Volume 2*. The University of Chicago Press, Chicago.

Miller, G. A. (1956). The magical number seven plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97.

Miller, G. A. (1962). Some psychological studies of grammar. *American Psychologist*, 17:748–762.

Miller, G. A. & Chomsky, N. (1963). Finitary models of language users. In Luce, D. R., Bush, R. R., & Galanter, E., editors, *Handbook of Mathematical Psychology*, volume II. John Wiley, New York.

Miller, G. A. & Isard, S. (1964). Free recall of self-embedded English sentences. *Information and Control*, 7:292–303.

Milne, R. W. (1982). Predicting garden path sentences. *Cognitive Science*, 6:349–373.

Minksy, M., editor (1967). *Computation: Finite and Infinite Machines*. Prentice-Hall, Englewood-Cliffs, NJ.

Mitchell, D. (1987). Lexical guidance in human parsing: Locus and processing characteristics. In Coltheart, M., editor, *Attention and Performance*, volume 12, pages 601–681. Lawrence Erlbaum Associates, Hillsdale, NJ.

Mitchell, D. C., Corley, M. M. B., & Garnham, A. (1992). Effects of context in human sentence parsing: Evidence against a discourse-based proposal mechanism. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18(1):69–88.

Mitchell, D. C., Cuetos, F., Corley, M. M. B., & Brysbaert, M. (1995). Exposure-based models of human parsing: Evidence for the use of coarse-grained (non-lexical) statistical records. *Journal of Psycholinguistic Research*, 24:469–488.

Newell, A. (1962). Some problems of basic organization in problem-solving programs. In Yovits, M. C., Jacobi, G. T., & Goldstein, G. D., editors, *Self Organizing Systems.* Spartan, Washington. D.C.

Newell, A. (1973). Production systems: Models of control structures. In Chase, W. G., editor, *Visual Information Processing.* Academic Press, New York.

Newell, A. (1980a). Physical symbol systems. *Cognitive Science*, 4:135–183. (Also available as CMU CSD Technical Report, Mar 1980; and in Norman, D. ed., *Perspectives in Cognitive Science*, Ablex, Norwood, NJ, 1981).

Newell, A. (1980b). Reasoning, problem solving and decision processes: The problem space as a fundamental category. In Nickerson, R., editor, *Attention and Performance VIII.* Erlbaum, Hillsdale, NJ.

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

Newell, A. (1992). Precis of *unified theories of cognition. Behavioral and Brain Sciences*, 15(3):425–492.

Newell, A. & Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In Anderson, J. R., editor, *Cognitive Skills and Their Acquisition.* Erlbaum, Hillsdale, NJ.

Newell, A. & Simon, H. A. (1972). *Human Problem Solving.* Prentice-Hall, Englewood Cliffs, NJ.

Perfetti, C. A. (1985). *Reading ability.* Oxford, New York.

Poizner, H., Bellugi, U., & Tweney, R. D. (1981). Processing of formational, semantic, and iconic information in American Sign Language. *Journal of Experimental Psychology: Human Perception and Performance*, 7(5):1146–1159.

Potter, M. C. (1982). Very short-term memory: In one eye and out the other. Paper presented at the 23rd Annual Meeting of the Psychonomic Society, Minneapolis.

Potter, M. C. (1993). Very short-term conceptual memory. *Memory & Cognition*, 21(2):156–161.

Pritchett, B. L. (1988). Garden path phenomena and the grammatical basis of language processing. *Language*, 64:539–576.

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

Pylyshyn, Z. W. (1984). *Computation and Cognition*. Bradford / MIT Press, Cambridge, MA.

Rayner, K., Garrod, S., & Perfetti, C. A. (1992). Discourse influences during parsing are delayed. *Cognition*, 45:109–139.

Reich, P. (1969). The finiteness of natural language. *Language*, 45(4):831–843.

Shulman, H. G. (1970). Similarity effects in short-term memory. *Psychological Bulletin*, 75(6):399–414.

Simon, H. A. (1974). How big is a chunk? *Science*, 183:482–488.

Singley, M. & Anderson, J. R. (1989). *The Transfer of Cognitive Skill*. Harvard University Press, Cambridge, MA.

Stabler, E. P. (1991). Avoiding the pedestrian's paradox. In Berwick, R. C., Abney, S. P., & Tenny, C., editors, *Principle-based Parsing: Computation and Psycholinguistics*. Kluwer Academic, Dordrecht, The Netherlands.

Stabler, E. P. (1994a). The finite connectivity of linguistic structure. In Clifton, C., Frazier, L., & Rayner, K., editors, *Perspectives in Sentence Processing*. Lawrence Erlbaum, Hillsdale, NJ. (An earlier version of this paper appeared in UCLA Occasional Papers No.11.).

Stabler, E. P. (1994b). The finite connectivity of linguistic structure. Unpublished manuscript, University of California, Los Angeles.

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

Tanenhaus, M. K., Carlson, G., & Trueswell, J. C. (1989). The role of thematic structures in interpretation and parsing. *Language and Cognitive Processes*, 4:SI 211–234.

Thibadeau, R., Just, M. A., & Carpenter, P. A. (1982). A model of the time course and content of reading. *Cognitive Science*, 6:157–203.

Tulving, E. (1983). *Elements of Episodic Memory.* Oxford University Press, New York.

Waugh, N. C. & Norman, D. A. (1965). Primary memory. *Psychological Review*, 72(2):89–104.

Wickelgren, W. A. (1965). Acoustic similarity and retroactive interference in short-term memory. *Journal of Verbal Learning and Verbal Behavior*, 4:53–61.