An Architecturally-based Theory of Human Sentence Comprehension

Richard Lawrence Lewis

December 18, 1993

CMU-CS-93-226

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:

Jill Fain Lehman, Chair
Jaime Carbonell, Computer Science Department
Marcel Just, Department of Psychology
Bradley Pritchett, Department of Philosophy

Allen Newell, Former Chair

This research was sponsored in part by the Wright Laboratory, Aeronautical Systems Center, Air Force Materiel Command, USAF, and the Advanced Research Projects Agency (ARPA) under grant number F33615-93-1-1330, and in part by a National Science Foundation fellowship.

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

Keywords: artificial intelligence, cognitive simulation, natural language processing, parsing and understanding, Soar

Abstract

This thesis presents NL-Soar, a detailed computational model of human sentence comprehension that accounts for a broad range of psycholinguistic phenomena. NL-Soar provides in-depth accounts of structural ambiguity resolution, garden path effects, unproblematic ambiguities, parsing breakdown on difficult embeddings, acceptable embeddings, immediacy of interpretation, and the time course of comprehension. The model explains a variety of both modular and interactive effects, and shows how learning can affect ambiguity resolution behavior. In addition to accounting for the qualitative phenomena surrounding parsing breakdown and garden path effects, NL-Soar explains a wide range of contrasts between garden paths and unproblematic ambiguities, and difficult and acceptable embeddings: the theory has been applied in detail to over 100 types of structures representing these contrasts, with a success rate of about 90%. The account of real-time immediacy includes predictions about the time course of comprehension and a zero-parameter prediction about the average rate of skilled comprehension. Finally, the theory has been successfully applied to a suggestive range of cross-linguistic examples, including constructions from head-final languages such as Japanese.

NL-Soar is based on the Soar theory of cognitive architecture, which provides the underlying control structure, memory structures, and learning mechanism. The basic principles of NL-Soar are a result of applying these architectural mechanisms to the task of efficiently comprehending language in real-time. Soar is more than an implementation language for the system: it plays a central theoretical role and accounts for many of the model's novel empirical predictions.

To my parents

and

the memory of Allen Newell

Contents

List of Tables

List of Figures

Acknowledgments

I must first acknowledge my mentor and friend for five years, Allen Newell. Though Allen did not live to see the completion of this thesis, if there is one hope I have for this work, it is that his influence would still be seen clearly in it. This thesis is not only a product of his scientific ideas, but his endless patience with me as well. Working with Allen was exciting from day one, and the excitement never wore off. I am deeply thankful for the time I was given to spend with him, and dedicate this thesis to his memory.

Jill Lehman's role as colleague and collaborator on the project has been indispensable. Jill is a student of language in a way that I—or Allen for that matter—have never been. Jill also had the unenviable task of assuming the role of my faculty advisor after Allen died. Had I been in her shoes, I wouldn't have wanted to do it, but I'm glad she was there.

I was delighted to be able to interest Jaime Carbonell, Marcel Just, and Brad Pritchett in this work, enough to serve on my thesis committee. Through their feedback, and their own research, they have improved the quality of this thesis. Brad in particular has had a major impact on this work. His gracious lending of linguistic expertise over the past two years helped push the model in important new directions.

Over the years a number of people have been associated with the NL-Soar group, and have made working on the project great fun. Besides Allen and Jill, Scott Huffman, Greg Nelson, Robert Rubinoff, and Shirley Tessler have all contributed in various ways. NL-Soar got its start before I showed up as a grad student: Gregg Yost worked with Allen on NL-Soar in preparation for Allen's William James lectures. Gregg was a great help early on as I was trying to figure out how Soar worked.

The local Soar community at CMU provided a terrific environment in which to do cognitive science. The larger Soar community, from Los Angeles to Groningen, has also been a continuous source of ideas and encouragement (not to mention travel opportunities!). Although we've never been able to deliver that elusive NL module that the community wants, it's been great to have a number of people around the world actually care about how the work is proceeding.

I knew almost immediately that the Computer Science Department at Carnegie Mellon was the place I wanted to spend my years as a graduate student, and never once regretted the decision to come here. Although the department continues to experience growth pains, it has been hard for me to imagine a better environment. Of course, having world-class psychology and computational linguistics programs across the lawn doesn't hurt either.

(Stepping back now and looking at all of these communities—from the local group to the global group—it's amazing to consider that in each case Allen played a significant role in creating and sustaining them.)

I've been richly blessed for a long time with teachers and professors who not only put in the time to give me a quality education, but took a personal interest in me (perhaps on many occasion I gave them little choice. . .), challenging and encouraging every step of the way. It is with great pleasure that I thank Dan Buckman, Fernando Gomez, Charlie Hughes, Mike Moshell, Odell Payne, Chuck Phelps, and Sarah Jane Turner. It is by no means overstatement to say that without them I would not be writing these acknowledgments.

What I value most about the last six and a half years in Pittsburgh are the friends I've made. Coming into work each day is not only fun, but educational, if you have the officemates I've had: Jim Aspnes, Stewart Clamen, Bob Doorenbos, Jade Goldstein, and Aarti Gupta. I'm going to miss them a great deal. Anurag Acharya, Tony Simon, Milind Tambe, and Erik Altmann, among others made life inside and outside the university more enjoyable (if not for Erik I might still be taking the theory qualifier). It's just not the same watching Seinfeld or hockey without Alonso and Risi Vera, and lots of pizza. It's been a delight to see Alonso and Risi grow together and I'm thankful for the time I got to spend with them. I also want to thank Noel for always making me feel welcome (where will I wash and wax my car now?)

The Squirrel Hill Christian Church took me in with open arms when I first came to Pittsburgh, and gave me a home-away-from-home for the next few years. I'll always be thankful for their prayers and challenges during that time. I'm also going to miss the Allegheny Center Alliance Church, especially the Snider's cell group. That group has played a greater role in my spiritual growth than they may ever know. Their prayers helped get me through the final months working on the thesis. And I know Mrs. Farmer's unfailing prayers were always heard from McKeesport.

One of my regrets in leaving Pittsburgh is that I won't get to see Sam and Sherry and the whole "Brunsvold" clan. I knew I could always count on Sam to be there through difficult times and joyful times. He made a significant impact on my life not only personally, but through his dedication to the campus ministry.

One of the *joys* of leaving Pittsburgh is that I'll be closer to Thad and Norma Polk and my new goddaughter Sarah! They have so enriched my life I can't believe how fortunate I've been that I got a chance to meet them in grad school. I have asked more of Thad than anyone should have to put up with. But I'll never forget all the fun we had while we were at CMU together.

My debts to friends (which is by now reaching embarrassing proportions) hardly stops in Pittsburgh. Seth Vanderdrift and Susan Smith have always been there over the past 15 years, even though we were separated by a thousand miles during the last six. I can't imagine what getting through school would have been like without them.

Finally, I want to thank my family, and especially my parents. I continually thank God for their unfailing encouragement, support, example, and love. This thesis is dedicated to them as well.

Chapter 1

Introduction

> *Now, the overwhelmingly puzzling problem about sentence comprehension is how people manage to do it so fast.*
>
> — Janet Fodor, Jerry Fodor, and Merrill Garrett (1975)

FODOR, FODOR, AND GARRETT certainly had it right. The ability to comprehend language in real time is one of the most complex and impressive of human cognitive skills. Equally impressive is the staggering amount of scientific effort that has been devoted to exploring the processes of comprehension. Few topics engage so many disciplines within cognitive science.

Over the past three decades, psychologists have uncovered regularities about aspects of comprehension ranging from lexical access to memory for text. Although many theories have been proposed to explain these regularities, most address a small set of phenomena, and only a few take the form of complete computational models. In artificial intelligence, there has been more concern for building processing models with increasing functional coverage, but most complete NLP systems still do not model any appreciable set of psychological phenomena.

A notable exception is the READER model of Thibadeau, Just, and Carpenter (1982), which is one of the earliest examples of a complete, functional comprehension system that attains some measure of psychological plausibility. The continued development of this theory (Just & Carpenter, 1992), along with some recent theories emerging from linguistics and computational linguistics (Gibson, 1991; Kempen & Vosse, 1989; Pritchett, 1988; Weinberg, 1993), indicates that unified computational accounts of certain aspects of sentence comprehension are within reach. Each of these theories addresses a significant range of phenomena with a single set of mechanisms or principles (a discussion of these and other theories appears in Chapters 2 and 9).

This thesis takes another significant step toward a unified theory of sentence comprehension by presenting a computational model, NL-Soar, that satisfies the following goals:

1. *Breadth.* The theory models a wider range of psychological phenomena than has previously been given a cohesive account.

2. *Depth.* The theory models the phenomena with a depth matching or exceeding the current best theories for those phenomena.
3. *Architectural basis.* The theory is embedded in an independently motivated theory of the cognitive architecture.
4. *Functionality.* The theory functions as a working comprehension system.

The remainder of this chapter elaborates these goals by providing an overview of the target phenomena, an explanation of what it means for the theory to be architecturally-based, and a preview of the theory and major results. The chapter concludes with a reader's guide to the remainder of the thesis.

1.1 The core sentence-level phenomena

NL-Soar addresses six kinds of phenomena that form a cluster of regularities at the *sentence level*. The phenomena are primarily about the on-line processes involved in piecing together the words in a sentence to form a meaning. Though NL-Soar necessarily embodies some plausible assumptions about lower-level processes such as lexical access, and higher level processes such as creating a long-term memory of the comprehended content, the theory does not yet model the phenomena at these levels in significant detail. However, the sentence-level processes and the phenomena surrounding them form an important core that must ultimately be addressed by any comprehension model. The phenomena are:

1. **Immediacy of interpretation and the time course of comprehension.** Our subjective experience is that we comprehend language incrementally, understanding each word as it is heard or read. As a hypothesis about the comprehension process, this has been advanced as the principle of *immediacy of interpretation* (Just & Carpenter, 1987), and much experimental evidence has accumulated in support of it. In general, immediacy holds for all levels of comprehension—syntactic parsing, semantic interpretation, and reference resolution. Furthermore, this immediate comprehension happens rapidly, at an average rate of ∼240 words per minute in skilled reading. Although the average time per word is ∼250 ms, eye fixation studies also reveal that fixations range from as little as 50 ms to 1000 ms or more.

2. **Ambiguity resolution.** When readers or listeners encounter an ambiguity, how do they decide which interpretation to give it? A theory of comprehension must specify what knowledge is brought to bear in resolving ambiguities, and how and when that knowledge is brought to bear. There are several kinds of ambiguities that arise in comprehension, ranging from lexical-semantic to referential, but here we primarily focus on *structural* ambiguity—alternative interpretations that arise because the partial utterance is consistent with multiple syntactic parses. (1) below gives a simple example:

   (1) The cop saw the dog with the binoculars.

Sentence (1) exhibits a structural ambiguity: the prepositional phrase *with the binoculars* can attach to *saw* or *dog*. General knowledge may prefer to interpret binoculars as the instrument of the seeing, but in certain specific contexts the binoculars may be associated with the dog.

The empirical evidence concerning the knowledge sources used to resolve ambiguities is mixed. Some studies have demonstrated that the semantic content of the sentence or the established discourse context can have an effect on the on-line resolution of local ambiguities. Other studies have shown the lack of such effects, demonstrating instead an apparent preference for one syntactic *structure* over another, independent of content or context.

3. **Garden path effects.** A garden path effect arises when a reader or listener attempts to comprehend a grammatical sentence with a local ambiguity, misinterprets the ambiguity, and is unable to recover the correct interpretation. The result is an impression that the sentence is ungrammatical or nonsensical. (2a) below, due to Bever (1970), is the classic example. *Raced* may be taken as the main verb of the sentence, or a relative clause modifying *horse*. The relative interpretation is globally correct. ((2b) has a parallel structure, but *driven* is unambiguous, so the garden path is avoided.)

   (2) (a) The horse raced past the barn fell.
   (b) The car driven past the station stopped.

   The subjective experience provides compelling linguistic evidence for the unacceptability of these sentences, but additional experimental evidence comes from reading times and grammaticality judgments. The reduced relative construction in (2a) is but one kind of garden path; a collection of 26 different types is presented in Chapter 2. Though the garden path effect has been well known since Bever's (1970) article, Pritchett (1988) was the first to deal in depth with the variety of constructions.

4. **Unproblematic ambiguities.** Some local ambiguities do not cause difficulty no matter which interpretation proves to be the globally correct one. Consider the pair of sentences in (3):

   (3) (a) I know John very well.
   (b) I know John went to the store.

   There is a local ambiguity at *John*, since it could either be the direct object of *know* or the subject of an incoming clause. Yet, regardless of the final outcome, the sentence causes no perceptible processing difficulty. There are a wide variety of constructions with unproblematic local ambiguities; Chapter 2 presents a collection of 31 different kinds. These constructions provide additional constraint for a theory intended to model garden path effects: the posited mechanism must be weak enough to predict difficulty on garden paths, but not so weak that it cannot process the unproblematic ambiguities.

5. **Parsing breakdown on difficult embeddings.** Some constructions *without* structural ambiguity are difficult for people to comprehend. The effect is similar to a garden path, but the source of the difficulty is not recovery from a misinterpretation; instead, it seems impossible to parse the sentence at all. Consider the following center-embedded sentence:

   (4) The dog that the man that the cat chased bit ran away.

   Such structures indicate that there is some kind of severe limit on the human capacity to handle certain kinds of embeddings. As with garden paths and unproblematic ambiguities, there exists a range of such structures, of which (4) is just one kind; Chapter 2 presents a collection of 17 different types. Gibson (1991) was the first to deal in depth with this variety.

6. **Acceptable embeddings.** Complementing the structures that cause parsing breakdown are those embeddings which do not cause such a breakdown. For example:

   (5) The dog that the man bit ran away.

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

   Sentence (5) shows that the structure in (4) becomes acceptable with one less embedded clause. (6) is an example of a fairly complex embedding (involving a sentential subject) which is nevertheless acceptable. A collection of 26 acceptable embeddings is presented in Chapter 2. Such structures constrain theories of parsing breakdown just as the unproblematic ambiguities constrain garden path theories.

The evidence for these phenomena comes from work in speech comprehension and reading, since they arise in both skills (in fact, cross-modal techniques are an important source of data). Although there are independent issues as well—for example, control of eye movements is not as critical in speech comprehension as in reading—a reasonable assumption is that a shared subset of the comprehension processes underly these shared phenomena. These common processes are what NL-Soar is intended to model.

1.2 Architectural basis

In the 1987 William James Lectures, Allen Newell issued a call for theoretical unification in psychology. Newell was concerned with the proliferation of microtheories in psychology, and offered the development of *cognitive architectures* as the path to a more cumulative science. A cognitive architecture is the fixed computational structure that supports cognition—it specifies the control structures, memories, and primitive processes underlying all cognitive behavior.

Though the notion of architecture was not new in 1987 (Newell & Simon, 1972; Card et al., 1983; Anderson, 1983; Pylyshyn, 1984), Newell established the feasibility of developing a single cognitive architecture that applies to a broad range of phenomena, sweeping the time scale of human behavior from immediate reaction tasks (hundreds of milliseconds) to long stretches of problem solving behavior (minutes or more). He also provided the necessary methodological tutelage, demonstrating how to take an architectural theory seriously and how to apply such a theory in a variety of ways.

The theory Newell used as an exemplar was Soar, an integrated architecture developed by Newell, John Laird, and Paul Rosenbloom (Laird et al., 1987). Soar is a problem space architecture that combines a long-term parallel recognition memory, a declarative working memory, an open, flexible control structure, and an automatic, continuous learning mechanism. The integration of problem solving and learning capabilities into general-purpose architectures has been a focus of recent research in artificial intelligence, and Soar is but one system among many (Laird, 1991). However, most AI architectures are not proposed as psychological theories (e.g., PRODIGY (Carbonell et al., 1989)). In psychology, Soar joins the ranks of other architectural theories of cognition, such as CAPS (Just & Carpenter, 1987) and ACT* (Anderson, 1983).

The comprehension theory presented in this thesis, NL-Soar, is built on the Soar architecture, and grows out of the theory Newell sketched in the lectures and later in his book (Newell, 1990). What does it mean for NL-Soar to be grounded in the Soar architecture? It means that NL-Soar specifies how the functional requirements of comprehension are computationally realized within the architectural mechanisms of Soar. NL-Soar takes these mechanisms as given—theoretical hypotheses independently motivated by other functional and empirical considerations. Grounding NL-Soar (or any other cognitive theory) architecturally also means it is possible to deal adequately with questions about what part of the computational model should be taken as carrying theoretical content, and what part is simply implementation detail (Pylyshyn, 1984; Newell, 1990).

Thus, the fact that NL-Soar is embedded in Soar is not an implementational sidenote to this thesis, but carries theoretical content. The total theory is a combination of the Soar architecture and the additional content specified by NL-Soar that realizes the functions of comprehension. Working within Soar also sheds new light on old issues such as modularity, and, as we shall see, raises completely novel issues as well. Chapters 3 and 9 will deal explicitly with the role of Soar in the theory and its predictions.

1.3 An implemented system

NL-Soar is an implemented natural language system running in Soar6 (which is implemented in C), and is being used by several researchers in the Soar community (see Chapter 9 for pointers to this other work). It consists of a set of 929 Soar productions, and more than doubles in size as a result of learning. All of the examples presented in the remaining chapters have been run through the system, unless explicitly noted otherwise.

1.4 Brief preview of theory and major results

The NL-Soar model realizes the functions of real-time comprehension with a small set of operators that perform syntactic parsing, semantic interpretation, and referential processing. The control structure is a mix of serial and parallel processing, and is open to modulation by multiple knowledge sources. The comprehension skill is held in a long term recognition memory that generates, selects, and applies comprehension operators. Most of the processing happens by immediate recognition (automatically), but NL-Soar can always break out into deliberate processing as necessary. As a result of Soar's chunking mechanism, comprehension is a continuously improving mix of deliberation and recognition behavior. (In fact, all of NL-Soar's recognitional operators arise via chunking.) Misinterpretations are corrected by an on-line repair process, which is constrained to ensure computational efficiency in the problem space search and recognition match. The partial syntactic structure in working memory is efficiently indexed in a manner that also ensures efficient recognition match.

The basic properties of NL-Soar and the Soar architecture interact to account for a wide range of sentence processing phenomena. The model predicts a variety of both interactive and modular ambiguity resolution effects. NL-Soar provides the first explicit model of deliberate recovery from garden path effects. In addition to accounting for a number of other qualitative phenomena surrounding garden paths and parsing breakdown, the model successfully makes detailed predictions on a collection of over 100 different garden path structures, unproblematic ambiguities, difficult embeddings, and acceptable embeddings. The collection is primarily English, but includes a range of cross-linguistic items, including head-final constructions. The model also makes a several quantitative predictions concerning the time course of comprehension, including the first zero-parameter prediction of comprehension rate.

1.5 Reader's guide to the thesis

The thesis has three major parts: phenomena, theory, and theory application. The phenomena are reviewed in Chapter 2, along with a description of previous theories. Chapter 3 presents the NL-Soar theory in detail. The application of the theory to the sentence-processing phenomena is presented in Chapters 4–8. Chapter 9 provides a summary of the model and the predictions, as well as a discussion of general theoretical issues and a few closely related models. For the one minute version of the thesis, look at Tables 9.1, 9.2, and 9.3.

Chapter 2 may be approached in several different ways. If your goal is to obtain just a brief overview of the phenomena themselves, the final section provides a summary. The tables in the middle of the chapter, listing the garden path and parsing breakdown examples, may also be helpful. If you are interested in the theoretical work, you can use the summary section to familiarize yourself with the basic phenomena, then read the theory discussions at the end of each major section.

Chapter 2

Human Sentence Comprehension: Phenomena and Previous Work

All theories live under the same sword of Damocles.
— Allen Newell

MODERN PSYCHOLINGUISTICS began with a concern for the relationship between sentence comprehension and the grammars of linguistic theory (Miller, 1962), and since then sentence processing has remained one of the most active areas in the field. Given the tremendous amount of accumulated work, it is impossible to provide a complete review here. Instead, the primary purpose of this chapter is threefold: 1) Establish the phenomena relevant to the model presented in Chapters 3–7; 2) Review previous theories proposed to explain these phenomena; and 3) Motivate the choice of phenomena by demonstrating how they provide great leverage in identifying the central mechanisms of sentence comprehension.

A section is devoted to each major phenomenon, and each section concludes with a review of the relevant theories. Each theory is discussed only with respect to the phenomenon of interest. This organization has the disadvantage that the discussion of any particular theory may be distributed across several sections. However, a few theories closely related to NL-Soar in goals and content (particularly, Gibson (1991), Just & Carpenter (1992), and Pritchett (1992)) will be examined again in Chapter 9.

2.1 The products of comprehension

Before examining the phenomena surrounding *how* sentence processing proceeds, we should first consider the functional requirements: *What* does comprehension need to do? Clark & Clark (1977) give two answers. In the narrow sense, which they term *construction*, comprehension serves to build a representation of the meaning of the linguistic input. The broader sense of comprehension includes *utilization*, which refers to what the listener or reader does with the meaning once it is grasped—e.g., store it in memory, believe it, do it (if it is an instruction), answer it (if it is a question). While keeping in mind the warnings of Clark and Clark that the processes of construction and utilization are often not clearly separable, the remainder of this chapter, and most of this thesis, treats comprehension in the narrow sense—as the construction of a meaning representation.

This section reviews evidence for three kinds of representations produced as final outputs or as intermediate by-products of sentence comprehension: a referential model, a context-independent semantic encoding, and a syntactic structure. The evidence will range from purely functional to empirical.

2.1.1 Referential representation

We have taken as given that comprehension produces a representation of the meaning of the linguistic input. But we need to be careful about what is meant by a representation of meaning, since philosophy and linguistics have traditionally used the term *meaning* in a restricted way.

Sense and reference

The standard conception of sentence meaning traces its roots back to a distinction made by Frege (1892). According to Frege, linguistic expressions have a *sense* and *reference.* The reference of an expression is what that expression denotes. The sense of an expression is the manner in which the reference is made. The distinction can be seen most easily in noun phrases:

(7) (a) The most famous professor at Carnegie Mellon
(b) Herbert Simon

Both expressions refer to the same Nobel Laureate, but in a different manner. Because their denotation is the same, the expressions can be interchanged in some sentences without affecting the truth value of the sentences:

(8) (a) Alonso had lunch with Herbert Simon.
(b) Alonso had lunch with the most famous professor at Carnegie Mellon.

However, Frege pointed out that a purely denotational account of sentence meaning is unworkable. Consider

(9) Bill thinks he is the most famous professor at Carnegie Mellon.

Sentence (9) clearly does not mean that Bill thinks he is Herbert Simon. The two statements are logically independent.

The notions of sense and reference have evolved into the more precise concepts of *intension* and *extension* in formal semantics (Lewis, 1972). In a semantics based on possible worlds (or states of affairs) the intension of a predicate (say *red*, or *Herbert Simon*) is a function from possible worlds to sets of objects. The extension of a predicate in a particular world is the set of objects in that world which are selected by the predicate; the objects are the result of applying the intensional function to the given world. Similarly, the intension of a sentence is a function from possible worlds to truth values. In other words, the intension is what determines whether the sentence is true in a particular situation. The extension of a sentence is then usually taken to be a truth value.

Given these definitions, the traditional view in philosophy and linguistics is that the *meaning* of a sentence corresponds to its intension, or truth conditions (Lewis, 1972). Yet it is clear from examples such as (8b) that meaning in this sense is often *not* all that is important. Depending on the listener's knowledge, what is being communicated in (8b) is not just that Alonso had lunch with the most famous professor at CMU *whoever that is*, but that Alonso had lunch with *Herbert Simon*. This leads to our first functional requirement for comprehension:

> Comprehension builds a *referential representation* which contains information about the *particular referents* of the discourse in the particular situation described by the discourse—not just the set of truth-conditions of the constituent sentences.

From an AI point of view, this is a somewhat obvious assertion to make, because the usefulness of a referential representation has been established since SHRDLU (Winograd, 1972). Nevertheless, this functional requirement was not always appreciated in psychology[1].

Work in psycholinguistics has led to an additional claim about the referential representation:

> The referential representation is what is retained as the primary and final product of comprehension.

A form of this claim first showed up first as the *constructive* theory of Bransford, Barclay, and Franks (1972). They proposed that comprehension constructs a representation of the described situation that integrates the information explicit in the input along with inferred or background information. The result is a single coherent representation of the content. In their experiments, subjects read short paragraphs describing simple configurations of objects, such as (10):

(10) Three turtles rested on a floating log, and a fish swam beneath them.

On subsequent recognition tests, they found that subjects could not accurately distinguish between sentences that were present in the paragraph and those that were merely consistent with the situation described by the sentences. What was retained was not the particular form in which the information was presented, or a set of sentence meanings, but a referential

[1] See, for example, the comprehension theory of Kintsch & van Dijk T. A. (1978), which did not contain a referential representation, and its subsequent evolution into a model in which such a representation plays an important role (Van Dijk & Kintsch, 1983)

representation of the described situation. In similar work, several studies have shown that subjects often confuse different NPs that refer to the same individual, indicating again that an *extensional* representation is retained (Anderson & Bower, 1973; Anderson, 1973; Garnham, 1987).

The form of representation: models vs. logics

The claim that comprehension produces a referential representation is a knowledge-level[2] claim. It asserts that information is encoded about the particular referents in the discourse, possibly along with previously known or inferred information. But it does not say *how* that information is encoded.

One possibility is that the representation is a logic of some kind. For example, if the predicate calculus is used, then knowledge about the referents is encoded as a set of first order statements, and reasoned with by applying the inferential rules of the calculus. The way the statements represent is defined by the model-theoretic semantics of the logic: the statements represent a particular situation if that situation is a *model* of the statements (Ebbinghaus et al., 1984). This correspondence is indirect. The elements of the representation—the variables, connectives, quantifiers, etc.—do not necessarily map directly onto the domain being represented.

Alternatively, there can be a *direct correspondence* between parts of the representation and parts of the represented situation. Individuals, properties, and relations in the representation can map directly onto individuals, properties, and relations in the domain. Such a representation is not a logic; Johnson-Laird (1983) calls it a *mental model*.

Although the comprehension experiments discussed earlier establish the reality of a referential representation, they cannot help settle issues about the nature of that representation. To distinguish logic and mental models, what is needed is a study of tasks that tap more directly into those aspects of logic that violate structure correspondence, such as quantification or disjunction. Over the past decade or more, Johnson-Laird and his colleagues have carried out just such a research program, studying subjects performing explicit reasoning tasks such as categorical syllogisms. These tasks are relevant to the issue of what comprehension produces because they involve the comprehension of premises presented in natural language.

Johnson-Laird's conclusion is that humans are *not* using a logic-based representation (Johnson-Laird, 1983; Johnson-Laird, 1988). The conclusion is based primarily on the pattern of errors that subjects produce, which are not consistent with a proof-based account of their reasoning. Instead, the results are consistent with a representational theory that can be summarized (in part) as follows:

> Comprehension builds a mental model of the linguistic input, which is a referential representation of a *particular situation* that is consistent with the input.

[2] To describe a system at the knowledge level is to describe the behavior of the system purely in terms of knowledge and goals, without specifying the underlying mechanisms that represent the knowledge and bring it to bear. A knowledge level claim is a claim about content, but not about how the content is encoded. See (Newell, 1990) or (Newell, 1982) for a full discussion.

> The elements of the representation correspond one-to-one to elements of the situation. Comprehension typically produces just one model, even when multiple models are possible.

The concept of a referential representation that takes the form of a mental model is now established in psychology and psycholinguistics. It plays a central role in most comprehension theories (for example, the situation model of (Van Dijk & Kintsch, 1983) and the referential representation of (Just & Carpenter, 1987)). It continues to receive empirical support from a range of work in psycholinguistics and reasoning (Bower & Morrow, 1990; Johnson-Laird & Byrne, 1991; Polk, 1992).

2.1.2 Semantic representation

Comprehension must also produce a reference-independent semantic representation that corresponds to sense or intension. Below we consider the functional and empirical evidence for such a representation.

There are two kinds of functional requirements for an independent semantic representation. First, uncovering the semantics of a sentence is a necessary step to producing the referential representation. The referents of expressions such as *the most famous professor at CMU* in sentence (8b) depends on the underlying semantics of the expression, not its particular surface form. Furthermore, there must be some capability of creating an initial representation of a referent when it is introduced for the first time. This capability is effectively a process of producing a reference-independent representation (because it cannot depend on first retrieving the referent).

The second functional requirement for a semantic representation is exemplified by Fregean examples such as (9). Such examples establish that the sense of an expression is sometimes independently needed in order to understand the expression. One-anaphora provides another good example of this requirement. Consider (11), adapted from (Allen, 1987):

(11) (a) Book a seat for Cynthia on the 6pm flight to Orlando.
(b) Book one for Katy, too.

Clearly, *one* in (11b) does not refer to the seat booked for Katy, but rather a seat that is identified by combining part of the description of the seat in (11a) with part of the description in (11b). There must be at least a temporary reference-independent memory of the first sentence if *one* is to be interpreted correctly in the second sentence.

There is also some empirical evidence for an independent semantic representation. Mani & Johnson-Laird (1982) performed experiments that tested memory for verbally presented spatial layouts (*The spoon is to the left of the knife. The plate is to the right of the knife.*, etc.). There were two kinds of descriptions: *determinate* descriptions were consistent with only one particular layout, and *indeterminate* descriptions were consistent with two distinct layouts. Mani and Johnson-Laird assumed that subjects attempt to construct a mental model of the described layout, and that two mental models are more difficult to construct and retain than one. The results were consistent with this assumption. Memory for the layout described by the determinate descriptions was better than for the indeterminate descriptions. What is relevant to the referential/semantic distinction, however, is that memory for the verbatim form of the description was better for the *indeterminate* descriptions. This suggests that the subjects retained a superficial semantic representation (which is naturally closer to the surface form of the input) when the indeterminacy of the input made building a mental model too difficult. As Mani and Johnson-Laird point out, it is difficult to make sense of such a cross-over effect without appeal to two different kinds of representations. A similar effect was found in an analysis of recognition memory for a programmer's manual (Schmalhofer & Glavanov, 1986).

The assumption that comprehension produces a semantic representation is widely adopted in AI and psychology. For example, it appears as the *logical form* of Allen (1987), the *propositions* of Johnson-Laird (1983), the *semantic representation* of Just & Carpenter (1987), the case frames of (), and the *propositional textbase* of Van Dijk & Kintsch (1983)[3]. All these representations have in common the central property that they encode the sense of linguistic expressions in some intermediate representation prior to computing reference.

2.1.3 Syntactic representation

In this section we consider the evidence for a syntactic structure produced as an intermediate product of comprehension. The evidence will be strictly functional, since there are still no empirical results universally accepted as indicating the existence of such a structure. The claim is not simply that syntax is required to comprehend language; that much is certainly universally accepted and almost tautological. Rather, the issue is whether or not a separate syntactic representation is computed and at least temporarily maintained during comprehension.

Making syntactic discriminations is necessary to arrive at the correct meaning. Each aspect of syntax, such as number agreement, potentially eliminates some semantic ambiguity. To make these discriminations, comprehension must maintain some local syntactic state—the process cannot just be syntactically driven and leave behind nonsyntactic representations. Two simple examples will illustrate the point:

(12) You are going to the store.

(13) John saw him in the mirror.

[3] I have avoided using *propositional* to refer to this level of representation, since the term is used in different ways in psychology. For some theorists, it is a content-independent format that can be used to encode any level of representation (e.g., (Just & Carpenter, 1987)); others reserve the term to refer only to the level of representation prior to reference resolution (Johnson-Laird, 1983). Rather than risk this confusion, I will continue to refer to the *semantic representation*, which places the focus on the content of the representation, rather than its form.

In (12), the syntactic number of the subject of the sentence (in this case, *you*) must be represented so that it can be checked against the number of the verb when it arrives. If the number feature is not independently represented, then every possible lexical subject/verb pair that agrees (e.g, *you/are*) must be stored, and the check performed by lookup. That the feature is not purely semantic is apparent from the number associated with the English second person pronoun *you*: the syntactic number is plural whether the pronoun refers to an individual or a set (**John, you is going to the store.*) In (13), the syntactic structure of the initial part of the clause (*John* in subject position) must be represented so that the proper reference for *him* can be computed. In this case, *him* cannot refer to *Jim*, otherwise the reflexive form (*himself)* must be used. The constraints on the distribution of reflexives is a function of syntactic, not semantic structure (e.g., (Chomsky, 1965; Chomsky, 1973)).

Despite these functional considerations, some scientists in both AI and psychology have argued that a syntactic representation is not needed. In AI, this claim is best represented by the work of Schank and colleagues (Schank & Riesbeck, 1981). They constructed a number of systems that parsed directly into a semantic representation (conceptual dependency) without an intermediate syntactic structure. In psychology, Johnson-Laird (1983) proposed that human comprehension is largely syntactically driven, but does not create syntactic structure.

However, no system built on such principles has managed to achieve anywhere near the structural coverage of systems that do use syntactic representations. The syntax-lean systems do, of course, embody some syntactic knowledge, but it primarily consists of ordering constraints that do not require explicit encodings to apply (Schank & Riesbeck, 1981). Some recent work that has emerged from the tradition of the Schank conceptual parsers does in fact handle more complex syntax, but, not surprisingly, these systems *do* make use of explicit intermediate syntactic structures (Cardie & Lehnert, 1991). Thus, both functional considerations and actual system building practice provide evidence for the necessity of explicit syntactic encodings of some kind.

2.1.4 Summary

Table 2.1 summarizes the three representations and the evidence for them. Such a brief characterization abstracts away from an enormous number of issues—for example, the entire field of syntax in linguistics—but it serves to establish the basic character of the output of comprehension and provide the necessary foundation for discussing the processing phenomena and the model presented in Chapter 3.

2.2 Immediacy and the time course of comprehension

We seem to comprehend speech on a word-by-word basis, as quickly as we hear it. We read at even more rapid rates—$\sim$240 words per minute (Just & Carpenter, 1987). The phenomenology has much objective empirical support. Carpenter and Just call this rapid incremental processing *immediacy of interpretation*: syntactic, semantic and referential

TABLE 2.1: The products of comprehension.

TYPE	NATURE OF REP'N	EVIDENCE
Referential	Organized around referents of the text, may not retain intensional distinctions. Form is a model which represents one particular situation consistent with text. Aspects of model correspond directly to aspects of situation.	Functionally required. Linguistic memory does not always reliably distinguish expressions that actually occurred from referentially or inferentially equivalent expressions.
Semantic	Represents sense of expressions (independent of reference)	Functionally required. Under certain conditions the semantic representation is retained rather than the referential representation.
Syntactic	Explicit encoding of intermediate syntactic structure.	Functionally required. Practice in building working NLP systems reveals this requirement.

processing follow immediately on the heels of each incoming word, and are rapidly completed in most cases. This section reviews some of the evidence for the immediacy of producing the three representations identified in §2.1.

2.2.1 Immediacy of syntactic parsing

Syntactic structure is computed incrementally—an incoming word is integrated immediately into a partial syntactic structure. The speech shadowing experiments of Marslen-Wilson (1973, 1975) provide striking evidence for the immediacy of syntactic processing. The subjects' task is to repeat back speech as soon as they hear it. The most practiced subjects shadowed at latencies of 350 ms or less. When subjects made errors by adding or changing words, the errors were consistent with the preceding syntactic context well over 80% of the time. Furthermore, this consistency with syntax was just as likely to occur at latencies of 250 ms (the most rapid shadowers) as it was at 600–1000 ms. These experiments indicate that syntactic processing of a word occurs within at least a few hundred milliseconds of hearing the word, otherwise the syntactic context would not be available for generating the next shadowed word so quickly.

The bulk of evidence for syntactic immediacy comes from tracking eye movements during reading. A number of studies have shown that eye fixations are of longer duration on those parts of a sentence that are syntactically difficult or anomalous (e.g., (Carpenter & Daneman, 1981; Frazier & Rayner, 1982)). These results suggest that subjects are attempting to syntactically parse material as soon as it is fixated. In fact nearly all of the vast set of experiments exploring structural ambiguity resolution (§2.3) provide evidence for syntactic immediacy.

Limits to syntactic immediacy?

While the general claim is uncontroversial, there is still some debate over whether *all* syntactic information becomes available on a word-by-word basis. If there is evidence that some aspects of syntax are delayed in their application relative to others, then that may provide evidence for independent modules within the grammatical knowledge base (Fodor, 1988)[4]. It would also weaken the general claim for syntactic immediacy. The range of possible hypotheses about delayed application of syntax is as rich as modern syntactic theory. Below we will consider just the most important and well-investigated of these claims: that verb argument structure (or subcategorization information) is not immediately available to guide the initial parse.

In a self-paced reading study, Mitchell (1987) presented subjects with material such as:

(14) (a) After the audience had applauded the actors sat down for a drink.
(b) After the audience had departed the actors sat down for a drink.

Mitchell manipulated the transitivity of the initial verb (*applauded/departed)* to see what affect this had on parsing the immediately following NP (*the actors)*. Mitchell found that subjects took the immediately following NP as a direct object whether the verb was transitive (14a) or obligatorily intransitive (14b).

However, truly obligatorily intransitive verbs are hard to come by, as example (14b) from Mitchell's material demonstrates: *departed* is not in fact obligatorily intransitive:

(15) The actors departed the stage.

The inadequacy of the material therefore makes the results suspect; this problem has plagued other studies purporting to show the same effect (Ferreira & Henderson, 1990).

On the other hand, there is positive evidence that subcategorization information *is* used immediately in parsing. As Pritchett (1992) points out, the garden path effect in (14) can be avoided by using a properly intransitive verb:

(16) While Mary slept a sock fell on the floor.

Such a contrast is difficult to explain if subcategorization information is delayed in parsing.

Tanenhaus and Carlson and colleagues (1989) have provided a great deal of empirical support for the immediate use of lexical information in parsing. They studied filler gap sentences:

(17) (a) The district attorney found out which witness the reporter asked ␣ anxiously about ␣.
(b) The district attorney found out which church the reporter asked ␣ anxiously about ␣.

[4] Failure to find such delays, however, does not mitigate against a modular grammatical theory.

(c) The physical therapist wasn’t sure which bed the orderly hurried rapidly toward ␣.

Subjects produced longer reading times at potential gap sites (marked with ␣) of transitive verbs (*asked*) when the fillers were semantically anomalous (17b), compared to semantically plausible cases (17a). But the long times could be eliminated by using verbs with an intransitive preference (17c). In those cases, subjects apparently did not attempt to associate the filler with the verb, indicating immediate use of verb argument structure.

2.2.2 Immediacy of semantic interpretation

Sentence meaning is computed on a word-by-word basis—an incoming word is integrated immediately into a partial semantic representation. The speech shadowing experiments that support syntactic immediacy also provide evidence for immediacy of semantic interpretation. Just as subjects’ errors were generally syntactically consistent with the preceding context, they were semantically consistent as well. The experiment reported in (Marslen-Wilson, 1975) factors out the contribution of semantics to the shadowing errors, clearly demonstrating that the effects are not purely syntactic.

Important evidence also comes from eye movement studies. For example, Carpenter & Just (1983) found that semantic context and meaning frequency affect the time that readers fixate on a polysemous word. Readers spent less time on words when the high frequency meaning was consistent with the context, indicating that readers are attempting to integrate the incoming word into the meaning of the sentence as it is being read. The cross-modal priming studies of lexical access also demonstrate rapid semantic interpretation. Priming effects due to the contextually inappropriate sense of a polysemous word disappear after a few hundred milliseconds (e.g., (Swinney, 1979)).

The gap-filling studies of Tanenhaus & Carlson (1989) cited above provide further evidence for semantic immediacy, since the effect of an implausible filler shows up immediately. This result was replicated using event-related brain potentials (Tanenhaus et al., 1990). The relevant finding is that the pattern of brain activity associated with semantic anomalies occurs about 400 ms after encountering the anomalous part of the sentence.

Limits to semantic immediacy?

Psycholinguists arguing for syntactic modularity have generated many results which might be interpreted to favor a model in which semantic interpretation lags significantly (at least a word or more) behind syntactic processing (see the studies referred to in §2.3. However, these experiments deal exclusively with the application of certain kinds of knowledge to resolve syntactic ambiguity. There has been *no* direct evidence showing that semantic interpretation is not happening immediately. The only issue raised by these studies is whether semantic information is used immediately to guide the syntactic parsing. The two issues are logically independent.

2.2.3 Immediacy of referential processing

Constructing a mental model requires building representations of new entities in the discourse, establishing relations among them, and identifying existing individuals when they are referred to (reference resolution).

The referent of an anaphor (noun phrase, proper noun, or pronoun) is computed immediately upon encountering the anaphor. Eye movement studies by Carpenter & Just (1977) provide evidence for immediacy of pronoun resolution. The interesting effect was the pattern of regressions: subjects often moved their eyes from the pronoun itself to the antecedent in the preceding text. Other compelling evidence comes from the cross-modal experiments of Tyler & Marslen-Wilson (1982). Subjects heard short paragraphs ending with a sentence fragment (such as *He ran towards* . . .). The task was to name the visually presented continuation probe, which was a pronoun (either *him* or *her*). The probe was always syntactically correct, but depending on the referent of *him* or *her*, the probe was either appropriate or inappropriate with respect to the preceding context. There was a naming latency advantage for the contextually consistent probes, indicating a rapid completion of the resolution process.

Dell, McKoon, and Ratcliff (1983) present evidence for immediacy of noun phase resolution. They gave subjects texts to read like the following:

(18) The burglar surveyed the garage set back from the street. Several bottles of milk were piled at the curb. The banker and her husband were on vacation. The criminal slipped away from the streetlamp.

At a 250 ms offset from the relevant noun phrase (*criminal*), subjects were presented a word for a recognition task. Words related to the referent of the nounphrase were named faster than words that were not. For example, *garage* would be primed 250 ms after encountering *criminal*. The words were only related by the relations established in the text itself, as opposed to general knowledge. Furthermore, the referring expression was not the same for the two noun phrases (*criminal* vs. *burglar*), to ensure that the effect was due to referential processing and not something more superficial.

Limits to referential immediacy

While the evidence clearly supports immediate initiation of reference resolution, the data concerning the *completion* of the process is more complex (Sanford & Garrod, 1989). For both pronouns and noun phrase anaphors, there appear to be cases where the resolution process is not completed until well after the anaphor is initially encountered (Carpenter & Just, 1977; Duffy & Rayner, 1990; Greene, McKoon & Ratcliff, 1992). In some cases, the structure of the text itself makes it impossible to correctly identify the referent when the anaphor is encountered. But other cases appear to be related to the processing required to compute the referent.

In general, reference resolution and mental model construction may require an arbitrary amount of inference. There is clearly some limit to the processing that can happen in real-time comprehension, and this must be under deliberate control to some extent. These limits have been explored in a variety of experimental paradigms. The results of the Mani & Johnson-Laird (1982) study discussed above suggest that a mental model may not always be constructed or retained if the text is difficult. Swinney & Osterhout (1990) presents cross-modal priming evidence for a distinction between *perceptual* (automatic, on-line) vs. *cognitive* (deliberate, post-processing) inferences. McKoon & Ratcliff (1992) also provide evidence for a minimal amount of automatic inferencing during comprehension. In general, the content of the referential representation may depend on a number of variables, such as the amount of time available for comprehension, prior knowledge, attention, goals in reading, superficial features of the text, and so forth (Just & Carpenter, 1987; Oakhill, Garnham, & Vonk, 1989; Schmalhofer & Glavanov, 1986; Simon & Hayes, 1979). The control of inferences during comprehension is still an open research issue in AI as well.

The time course of processing

Studies of eye fixations durings skilled reading indicate that, on average, each word is processed in about 250 ms (Just & Carpenter, 1987). However, the amount of time spent on each word can vary greatly, from as little as 50 ms to 1000 ms or more. The time spent is a function of many features of the text, such as word frequency, syntactic complexity, familiarity with content, and ambiguity (e.g., (Just & Carpenter, 1987; Carpenter & Just, 1983; MacDonald et al., 1993).

2.2.4 Theories of immediacy

Although immediacy is a central tenet in a number of important comprehension theories, there are very few computational theories that actually model the time course of comprehension. This requires developing a comprehension theory within some computational architecture and giving a temporal interpretation to the processing primitives of the architecture.

READER (Thibadeau et al., 1982; Just & Carpenter, 1987) is the first example of a functionally complete model that accounts for the time course of comprehension in any significant way. READER is developed in the CAPS production system, which operates on continuous cycles of match and activation propagation. Thibadeau et al. show that by interpreting the number of processing cycles that CAPS takes per word as a measure of reading time per word, READER is able to provide a good account of the reading times of human subjects. There are two features of the model that contribute to the good fit. First, the lexicon in READER is constructed so that the initial activation levels of word senses is a function of the frequency of that word sense, so that low frequency senses require more cycle time to boost activation to threshold levels. Second, since READER embodies the basic principle of immediacy of interpretation, it spends longer on words that immediately trigger significant amounts of syntactic, semantic, or referential processing.

2.3 Structural ambiguity resolution

Natural language ambiguities may be classified along two independent dimensions: *content* and *scope*. Content identifies a particular aspect of some representational level—for example, lexical semantics or syntactic attachment. Scope refers to the span of input over which the ambiguity persists. A *global* ambiguity typically refers to an ambiguity that cannot be resolved by taking into account the entire sentence. A *local* ambiguity is a temporary ambiguity that will be resolved at some later point in the sentence. For example, (19) exhibits a global structural ambiguity at *with* (underlined). There is a choice at the syntactic level which cannot be resolved by the end of the sentence: the prepositional phrase may be attached to *cop* or *saw*.

(19) The man saw the cop with the binoculars.

(20) exhibits a local lexical ambiguity at *can* (auxiliary vs. main verb reading) which is resolved by the end of the sentence (in fact the next word).

(20) These factories can tuna very efficiently.

Lexical ambiguities such as (20) which are syntactic in nature effectively give rise to structural ambiguities. Not all lexical ambiguities are structural; ambiguities such as (21) are purely semantic:

(21) The old man's glasses were filled with sherry.

Both interpretations of *glasses* (drinking vs. optical) yield precisely the same syntactic structure.

The remainder of this section reviews some of the phenomena surrounding the resolution of local structural ambiguity. Lexical ambiguity will be considered only to the extent that it gives rise to structural ambiguity, as in (20). The central theoretical questions are: What knowledge sources are brought to bear in resolving the ambiguities, and how and when are these sources applied?

2.3.1 Structural preferences

Certain ambiguities have preferred interpretations that can be characterized in purely structural terms. For example, consider (22):

(22) Thad said that Tony flew to Atlanta yesterday.

There is a preference to associate *yesterday* with *flew* rather than *said*, though both interpretations are equally grammatical and plausible. Kimball (1973) attributes this to a preference of human parsing called *Right Association*:

(23) *Right Association:* Terminal symbols optimally associate to the lowest nonterminal node.

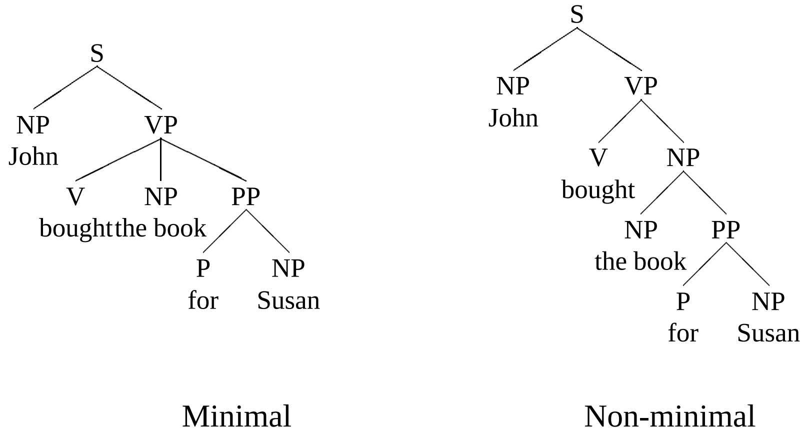


FIGURE 2.1: How to minimally attach a PP. The minimal attachment introduces the fewest number of new nodes.

Since the VP node headed by *flew* is lower than the one headed by *said*, the incoming material is attached to *flew*. Two other important preferences proposed in the literature are the *Canonical Sentoid Strategy* (Bever, 1970; Fodor et al., 1974) and *Minimal Attachment* (Frazier & Fodor, 1978):

(24) *Canonical Sentoid Strategy:* . . . whenever one encounters a surface sequence NP V (NP), assume that these items are, respectively, subject, verb, and object of a deep sentoid.

(25) *Minimal Attachment:* Each lexical item (or other node) is to be attached into the phrase marker with the fewest number of nonterminal nodes linking it with the nodes which are already present.

The Canonical Sentoid Strategy accounts for the bizarre interpretation of (26) (Pritchett, 1992):

(26) Japanese push bottles up Chinese.

The sequence *Japanese push* is interpreted as subject-verb, rather than as an NP. The prepositional phrase ambiguity in (27) illustrates how Minimal Attachment works:

(27) John bought the book for Susan.

Figure 2.1 gives the structures for the two possible attachments. Minimal Attachment selects the one with the fewer nodes, thus predicting the preferred attachment of *for Susan* to the VP node.

Table 2.2 lists a number of structural preferences that have been identified in the literature. They are listed here primarily as descriptions of phenomena, without any intention of denying them theoretical status. In §2.3.5 we will consider some of these as theoretical constructs.

TABLE 2.2: Some structural parsing preferences.

Canonical Sentoid Strategy (Bever, 1970; Fodor et al., 1974)	Interpret N-V-N string as subject-verb-object. Example: *Japanese push bottles up Chinese.*
Right Association (Kimball, 1973)	Attach to rightmost (lowest) open phrase. Example: *John said it would rain yesterday. Yesterday* attaches to *rain.*
Early Closure (Kimball, 1973)	Close phrases as soon as possible. Example: *They knew the girl was in the closet.* The S node is closed prematurely at *the girl*, accounting for processing difference with the unambiguous *They knew that the girl was in the closet.*
Minimal Attachment (Frazier & Fodor, 1978)	Attach with minimal number of nodes. Example: *John bought the dress for Susan. For Susan* attaches to *bought.*
A-over-A Early Closure (Church, 1980)	Given two phrases in the same category, the higher closes only when both are eligible for Kimball's Early Closure. Example: *I called the guy who smashed my car a rotten driver. Driver* may attach to *called* because the latter remains open.
Late Closure (Frazier & Rayner, 1982)	Delay closing constituents; prefer to attach new material to existing nodes. Example: *Since Jay jogs a mile seems like a short distance. A mile* attaches to *jogs* initially, causing a garden path.
Prefer Arguments (Abney, 1989)	Prefer argument (complement) attachments over non-argument (adjunct) attachments. Example: *The man expressed interest in the Volvo; in the Volvo* attaches to *interest* as an argument, rather than to *expressed* as a locative adjunct.

How robust are these preferences? Does human parsing always operate in accordance with some syntactic preference(s), or can these preferences be changed by semantic or pragmatic factors? Drawing in part on the hypotheses of Fodor (1983) and Forster (1979), many psychologists have proposed that syntactic processing is modular. The claim is that an autonomous syntactic module is responsible for structuring linguistic input, and this structuring is accomplished (initially, at least) without regard to non-syntactic information. Fodor (1983) calls this latter property, which is an essential feature of modularity, *information encapsulation.* Under the modularity view, the interesting issue is finding out what structural principles (perhaps of those listed in Table 2.2) govern the parser's operation at ambiguities, because these principles may reflect the basic nature of the syntactic processor.

A number of empirical studies have been carried out in support of the modularity hypothesis. Nearly all have the same basic structure:

- Subjects are presented with material that contains a local structural ambiguity.
- The manipulated variable is some nonsyntactic context prior to or including the local ambiguity.
- Readings times are recorded in the disambiguating region, as a measure of comprehension difficulty.
- If the measure of difficulty is independent of the nonsyntactic manipulation, this is taken to support a purely syntactic ambiguity resolution strategy, and a modular parsing architecture impervious to context.

For example, Ferreira & Clifton (1986) present evidence that ambiguous NP-V strings are interpreted as subject-verb (in accord with Minimal Attachment and Canonical Sentoid) regardless of the implausibility of the reading. Material included sentences such as:

(28) (a) The evidence examined by the lawyer turned out to be unreliable.
(b) The defendant examined by the lawyer turned out to be unreliable.

In (28a), the inanimacy of *evidence* makes it implausible that the evidence was doing the examining. Yet, increased reading times (over unambiguous controls) were detected in both cases in the disambiguating region, suggesting that subjects incorrectly interpreted the first verb as the matrix verb.

Table 2.3 summarizes some of the experiments supporting syntactic modularity generally (and Minimal Attachment specifically)[5]. The structure of the table is as follows: *Ambiguity* refers to the kind of structural ambiguity studied; *Manipulation* refers to the kind of nonsyntactic information that the ambiguity resolution was found to be insulated from; *Method* refers to the mode of presentation and behavioral measure.

The results of these studies should be interpreted with care (Tanenhaus et al., 1989; Tyler, 1989). Generalizing over these materials—for example, to conclude from the Ferreira & Clifton (1986) study that animacy never affects on-line ambiguity resolution—would be ill-advised, especially in light of the interactive results discussed in §2.3.3. Nevertheless, the studies do show at least that there are some combinations of subjects, ambiguity types, and nonsyntactic content that give rise to modular effects.

[5] A couple of well-known Minimal Attachment studies are not included here. The Frazier & Rayner (1982) study established a preference for Minimal Attachment, but did not explicitly examine any nonsyntactic factors. The Rayner et al. (1983) study manipulated plausibility, but the semantically disambiguating information came after the local ambiguity, so that information could not have been brought to bear on-line in any theory (except one involving lookahead).

The Britt et al. (1992) experiment which was intended to test for context effects on main verb/reduced relative ambiguities was not included in the table because the material did not actually seem to include the relevant contextual manipulation (changing the pool of potential referents for the subject NP), unlike their PP attachment experiment reported in the same paper; see Table 2.4.

TABLE 2.3: Some studies demonstrating modularity effects.

AMBIGUITY TYPE	MANIPULATION	METHOD
Main-verb/reduced relative	Animacy of subject	ET (Ferreira & Clifton, 1986; Just & Carpenter, 1992)
	Context (# of referents)	ET (Ferreira & Clifton, 1986); SPP (Ferreira & Clifton, 1986)
	Context (discourse focus)	ET (Rayner et al., 1992)
Complement/relative clause	Context (# of referents)	SPW (Mitchell et al., 1992)
PP attachment (arg/adjunct)	Context (# of referents)	ET (Ferreira & Clifton, 1986)
	Context (discourse focus)	ET (Rayner et al., 1992)

ET = eye tracking, SPW= self-paced word-by-word, SPP = self-paced phrase-by-phrase

How often will a parser guided by purely structural preference choose the correct interpretation? Though most preferences have been motivated by a narrow range of linguistic examples, there have been some recent analyses of natural corpora (Gibson & Pearlmutter, 1993; Hindle & Rooth, 1991; Hobbs & Bear, 1990; Whittemore & Ferrara, 1990). The study by Whittemore & Ferrara (1990) tests the predictions of several different attachment heuristics on 725 sentences containing prepositional phrase attachment ambiguities. No single heuristic they considered works better than 55% of the time. What works best is essentially a combination of lexically-specific information—especially argument structure—and right association. This basic pattern also held in other analyses (Gibson & Pearlmutter, 1993; Hindle & Rooth, 1991; Hobbs & Bear, 1990)[6].

Apart from the inadequacy of any single strategy, perhaps the most striking result of the Whitemore et al. study is the dismal performance of Minimal Attachment: it correctly predicts attachment only 36% of the time. These results do not in and of themselves falsify any structural preference as a theory of on-line parsing, because the data is based only on the final preferred structure. But it does make abundantly clear the need for a theory of *reanalysis* to complement these preferences, otherwise they risk grossly overpredicting garden path effects (see §2.4).

2.3.2 Lexical preferences

The specific lexical content of an ambiguous sentence can sometimes make a difference in its preferred interpretation. Ford, Bresnan, and Kaplan (1982) show that different words may differ in the arguments they prefer. The crucial motivating examples are minimally contrasting sentences such as (29):

[6] The percentage of attachments accounted for by a lexical preference/right association strategy was actually lower in the Gibson & Pearlmutter (1993) study than in the earlier studies. The authors attributed this to two factors: 1) The earlier studies focused exclusively on NP/VP attachment ambiguities, while their study examined NP/NP/NP attachment ambiguities. 2) In the case of the Whitemore et al. study, the domain was restricted, while the Gibson and Pearlmutter study used example taken from the Brown corpus.

(29) (a) The woman wanted the dress on that rack.
(b) The woman positioned the dress on that rack.

In (29a), the preferred interpretation associates *on that rack* with *dress*, but in (29b), *on that rack* is associated with *position*. Since the alternative structures for both sentences are putatively identical and only differ in the specific verb used, Ford et al. argue that this is evidence for a purely lexical preference. They established the phenomenon with an informal survey of subject intuitions about the meaning of sentences like (29). Some of the results of grammaticality judgment tasks reported in (Kurtzman, 1985) provide further evidence of lexical effects. The PP attachment studies mentioned above also indicate effects of lexical preferences, but not necessarily the specific kind proposed by Ford et al.: most of these effects may be due to simply to a preference for arguments over adjuncts.

Ford et al. note that context can apparently override lexical preferences:

(30) (a) When he arrived at our doorstep, I could see that Joe carried a package for Susan.
(b) Whenever she got tired, Joe carried a package for Susan.

Even in the absence of biasing context, lexical preferences must not be absolute, because as Gibson (1991) points out, counterexamples can be found:

(31) I wanted the dress for Susan.

Here, the PP does seem to attach to the verb, counter to the preference used to explain (29a).

2.3.3 Semantic and contextual effects

Locally ambiguous material is sometimes interpreted as a function of the local semantic content of the sentence itself, or the prior discourse context. Such effects are called *interactive* since they demonstrate the interaction of multiple knowledge sources in the comprehension process (Marslen-Wilson, 1975). For example, Tanenhaus et al. (1989) and Just & Carpenter (1992) present evidence suggesting that, contrary to what Ferreira & Clifton (1986) found, some subjects *do* in fact make rapid use of animacy information in resolving the local ambiguity of sentences like (28). Crain & Steedman (1985) produced similar results on a rapid grammaticality judgment task, showing that the plausibility of the grammatical reading affected the chances of sentences with reduced relative ambiguities being called grammatical.

Structural ambiguities may also be resolved by appeal to the current context. Tyler & Marslen-Wilson (1977) present striking evidence of the rapid effect context can have on syntactic processing (see also (Marslen-Wilson & Tyler, 1987)). Subjects heard sentence fragments like (32), ending with ambiguous strings (in italics).

(32) (a) If you walk too near the runway, *landing planes* . . .
(b) If you've been trained as a pilot, *landing planes* . . .

At the offset of the final word in the ambiguous phrase (*planes*), a probe word was visually presented. The word was a verb which was a continuation of the sentence. The subject's task was to name the verb. The contextually appropriate continuation is *are* for (32a) and *is* for (32b). Appropriate continuations had a naming latency advantage over inappropriate continuations, indicating that the context had a rapid effect on the initial analysis of the ambiguous string. Using a different technique (rapid grammaticality judgment), Crain & Steedman (1985) showed that the number of referents established in the context can affect the perceived grammaticality of a locally ambiguous sentence.

Demonstrations of contextual or semantic effects have sometimes been criticized for not being sensitive to on-line attachment choices (Clifton & Ferreira, 1989). The critics emphasize that what is at issue is not the final interpretation given an ambiguous string—no one denies that nonsyntactic information can ultimately have a tremendous influence—but the initial attachment choices made by the parser. While some experiments may be subject to this criticism, it should be clear from the examples above that there are demonstrations of interactive effects that use techniques sensitive to the immediate interpretation of ambiguities (e.g, the priming techniques of Marslen-Wilson, and the eye movement studies of Carpenter and Just). Even the studies which could arguably be insensitive to on-line processes (those employing rapid grammaticality judgments) produce results that are problematic to explain with a strongly modular theory (Altmann, 1988; Steedman & Altmann, 1989).

Interactive effects have now been demonstrated across a range of syntactic ambiguity types, knowledge sources, and experimental techniques. Table 2.4 summarizes some of these studies in the same format used in Table 2.3 to present the modular experiments.

2.3.4 Limited parallelism

Most of the discussion above has implicitly assumed that a single interpretation is selected at ambiguous points. However, it is possible that multiple interpretations might be computed and maintained in parallel, and this is an important theoretical and empirical issue.

In fact, a number of early studies that showed clear effects of ambiguity were taken to support a limited *multiple meanings* model, in which multiple meanings of an ambiguous phrase are computed and maintained until context selects one, or until the current clause is closed (Clark & Clark, 1977). For example, MacKay (1966) presented subjects with sentence fragments to complete, and discovered that ambiguous fragments took longer to complete than unambiguous fragments. Lackner & Garrett (1972) demonstrated an ambiguity effect in an interesting task where subjects were required to paraphrase a sentence heard in one ear, while ignoring a sentence heard in the other ear. The interpretation of an ambiguous sentence such as (33) could be influenced by an unattended biasing sentence such as (34):

(33) The spy put out the torch as our signal to attack.

(34) (a) The spy extinguished the torch in the window.
(b) The spy displayed the torch in the window.

TABLE 2.4: Some studies demonstrating interactive effects.

AMBIGUITY TYPE	MANIPULATION	METHOD
Main verb/reduced relative	Animacy of subject	ET (Just & Carpenter, 1992; Tanenhaus et al., 1989)
	Temporal context	ET (Trueswell & Tanenhaus, 1992)
	Plausibility	SPW (Pearlmutter & MacDonald, 1992), RGJ (Crain & Steedman, 1985)
PP attachment (NP or VP)	Context (# of referents)	WS (Altmann, 1987); ET (Britt et al., 1992), SPW (Britt et al., 1992); SPP (Britt et al., 1992)
	Semantic content of prior VP	SPW (Taraban & McClelland, 1988)
	Content of VP object	SPW (Taraban & McClelland, 1990)
Complement/relative clause	Context (# of referents)	RGJ (Crain & Steedman, 1985); WS (Altmann, 1987); ET (Altmann et al., 1992)
Adjectival/gerund	Content of initial phrase	LNC (Tyler & Marslen-Wilson, 1977; Marslen-Wilson & Tyler, 1987)
Subject/object	Syntactic context	RGJ (Warner & Glass, 1987)
	Semantic context	RGJ (Warner & Glass, 1987)

ET = eye tracking, LNC = lexical naming of continuations, RGJ = rapid grammaticality judgment, SPP= self-paced phrase-by-phrase, SPW = self-paced word-by-word, WS = whole sentence

In general, the closer to the ambiguity a technique probes, the more likely effects due to ambiguity will appear (Foss & Jenkins, 1973). Some studies demonstrated that ambiguity effects disappear altogether following clause boundaries, supporting the theory that multiple interpretations are maintained within clauses, and all but one is discarded at clause boundaries (Bever, Garrett, & Hurtig, 1973).

Most of these earlier studies were not focused exclusively on structural ambiguity, as the material in (33) illustrates. Furthermore, the studies often used non-trivial post-comprehension tasks (e.g., sentence completion or paraphrasing) that prevented direct assessment of the time course and nature of ambiguity resolution. In contrast, much subsequent work specifically addressing structural ambiguity with on-line techniques has yielded evidence consistent with the immediate selection of a single structure (see §2.3.1 and (Frazier, 1987)).

Researchers have recently turned back to trying to find direct evidence for structural parallelism. Kurtzman (1985) demonstrated clearly the complexity of the phenomena. Using rapid grammaticality judgments of sentence fragments, Kurtzman showed that multiple structural interpretations *may* be available at disambiguating points, but that their availability depends on a variety of syntactic and possibly pragmatic factors. Using syntactic and semantic priming techniques, Gorrell (1987), Hickok (1993), and Nicol & Pickering (1993) demonstrated that both structural interpretations of some ambiguous strings are available after the ambiguous point. MacDonald, Just, and Carpenter (1992) showed that subjects spend longer reading ambiguous regions than unambiguous controls, suggesting that multiple interpretations are being computed.

The overwhelmingly puzzling aspect of the Gorrell, Hickok, and Nicol and Pickering studies is that they provide evidence for the maintenance of the *unpreferred interpretation* of a sentence that causes a severe garden path. Consider the following example from Hickok:

(35) The psychologist told the wife that the man bumped that her car was stolen.

Hickok (and Nicol and Pickering) found evidence that people compute the relative clause interpretation for this sentence, even though it is precisely the unavailability of this interpretation at some level that causes the garden path. (The garden path status of this structure was clearly established with a separate grammaticality judgment experiment (Hickok, 1993), in which sentences like (35) were judged ungrammatical about 99% of the time.) Although all the researchers present some possible explanations for this result, no wholly satisfactory and coherent account has been developed.

2.3.5 Theories of ambiguity resolution

Strategy-based comprehension

Preferences such as Canonical Sentoid and Right Association, discussed in §2.3.1, were originally developed as part of a theoretical framework that might best be termed *strategy-based comprehension*. Strategies provided the first theoretical apparatus that separated the performance system from the competence grammar. As it became clear that the transformations of generative grammars in the 1960s did not correspond to processes of parsing and interpretation (e.g., (Fodor & Garrett, 1967)), psychologists turned to the perceptual mapping strategies to carry the full burden of explaining comprehension. Under this view, comprehension consists primarily of a collection of interacting syntactic and semantic heuristics that map surface form to some underlying structure (Bever, 1970; Kimball, 1973; Clark & Clark, 1977; Van Dijk & Kintsch, 1983). These heuristics were assumed to reflect basic cognitive or development constraints (Bever, 1970).

Though the strategy-based approach has been extremely influential, it has three fundamental problems:

1. Despite the general cognitive or developmental motivation for the approach, the specific strategies are ad hoc in nature.

2. A strategy-based theory often does not make clear predictions and is therefore difficult to test. The reason is that each strategy is not an absolute mapping rule, but a construct operating in concert with a heterogeneous collection of strategies. The interactions among these strategies were never well-specified, and no complete computational models were constructed. Furthermore, as Pritchett (1992) points out, the separation of the strategies from a competence grammar results in a curious situation: the strategies do not depend on any particular grammatical theory, but sometimes depend on specific grammatical rules.

3. When the structural strategies are formulated precisely and used individually to predict human preferences, a number of empirical problems arise in accounting for global as well as on-line preferences. This was demonstrated by the Whittemore & Ferrara (1990) study discussed earlier, as well as the interactive studies (Table 2.4). For detailed empirical critiques of several of the proposed strategies, see (Gibson, 1991) and (Pritchett, 1992).

The Sausage Machine and Minimal Attachment

The concern for the ad hoc nature of parsing strategies led to the attempt to more carefully derive the strategies from some underlying parsing architecture. The best known example is the Sausage Machine (Frazier & Fodor, 1978). The Sausage Machine is a two stage model. The first stage is the preliminary phrase packager (PPP, the Sausage Machine proper), which operates within the context of a restricted six-word window and assigns lexical and phrasal nodes. The second stage, the Sentence Structure Supervisor (SSS), sweeps along behind the PPP and structures the phrases into sentence nodes. Right Association is not a stipulated preference but emerges because the fixed window restricts available attachment sites. Any remaining ambiguities are resolved by the principle of Minimal Attachment (25).

The Sausage Machine and Minimal Attachment made two significant theoretical advances over the earlier strategy-based theories. First, they are more clearly motivated by computational considerations: Right Association emerges from the fixed window, which reduces the parsing search space, and Minimal Attachment is formulated to keep the short term memory load to a minimum. Second, Minimal Attachment applies to a wider range of cases than other strategies, which tend to be specific to particular structures.

The theoretical and empirical problems with the Sausage Machine and Minimal Attachment are well known (e.g., (Wanner, 1980; Abney, 1989; Gibson, 1991; Pritchett, 1992). Briefly, they include:

1. The inability of the six-word window to correctly predict parsing preferences on short sentences;

2. The inability to account for the on-line semantic and pragmatic effects identified in §2.3.3;

3. The grammatically suspect assumption that adjunction to NP uniformly introduces new nodes while adjunction to VP does not;

4. A range of false attachment predictions that derive partly from Minimal Attachment's insensitivity to argument structure;[7]

5. The vague appeal to computational justification—while an advance over earlier theories, MA was still developed in the absence of any precisely articulated assumptions about computational structures or processes, and in the absence of a model of comprehension that goes beyond parsing (Altmann, 1988).

Nevertheless, Minimal Attachment and the Sausage Machine set the agenda for much of the sentence processing work that followed. This work includes attempts to characterize attachment preferences in terms of alternative parsing architectures, such as ATNs (Wanner, 1980).

Generalized Theta Attachment

Recently, Pritchett (1988, 1992) has advocated a return to strongly grammatically-derived processing models of the kind that were abandoned shortly after their conception in the 1960s. Pritchett's theory of how parsing proceeds is captured in the following statement:

> (36) *Generalized Theta Attachment*[8] (GTA): Every principle of the Syntax attempts to be maximally satisfied at every point during processing.

"Every principle of the Syntax" refers to the principles of Government and Binding theory (Chomsky, 1981). To illustrate how GTA works, consider the following example:

> (37) I donated the gifts to the church . . .

The preferred attachment of *to the church* is as the second argument of *donate*, rather than as a modifier of *gifts*. GTA predicts this because the argument attachment maximally satisfies the *Theta Criterion* (Chomsky, 1981), which states, roughly, that each thematic role must be assigned to exactly one argument and each argument must bear exactly one thematic role.

GTA accounts for many of the same effects as Canonical Sentoid, Minimal Attachment, and Prefer Arguments but elegantly collapses these preferences into a single principle, without many of the linguistic and empirical difficulties of Minimal Attachment. Though GTA bears a family resemblance to theories such as Minimal Attachment, it differs from its immediate predecessors in two important ways. First, the theory derives its empirical predictions from a particular syntactic theory, namely GB. Second, the theory is not motivated by any extra-grammatical assumptions (apart from the GTA itself) such as short-term memory limitations or fixed windows.

[7] A problem related to the unusual assumptions about adjunction; but see Abney (1989) for an explanation of why modifying these assumptions leaves Minimal Attachment empirically inert.

[8] The name reflects the historical development of the theory from one concerned primarily with theta-marking to one that appeals to all syntactic principles equally. I prefer to think of it as *Greedy Government and Binding Parsing.*

There are two kinds of phenomena left unaccounted for by GTA. GTA does not predict Right Association effects. However, in this regard, the theory is no worse off than any other, because every previous theory must account for this preference by simply positing it as an additional principle[9]. The most serious issue for GTA is that (like other purely syntactic theories) it cannot account for the on-line semantic and contextual effects in §2.3.3.

Lexical preferences

Ford, Bresnan, and Kaplan (1982) proposed a theory of ambiguity resolution motivated by their data on lexical preferences (§2.3.2). The lexical theory posits that each lexical form of a given verb has an associated strength, and verbs differ in the strengths of their different forms. In the case of (38), the form of *want* that takes a single NP complement is strongest, while *positioned* prefers a double complement.

(38) (a) The woman wanted the dress on that rack.
(b) The woman positioned the dress on that rack.

There are two important issues in considering lexical preferences as a theory of ambiguity resolution:

1. How do lexical preferences affect the immediate, on-line attachment decisions?
2. How do lexical preferences interact with other knowledge sources to arrive at the initial or final interpretation?

The first issue is related to the general question of how lexical information affects on-line parsing (e.g., see the discussion of subcategorization in §2.2.1). The data presented in (Ford et al., 1982) does not address this issue since it concerns only the final interpretations given to globally ambiguous sentences (cf. the studies summarized in Table 2.3). As for the second issue, the authors' themselves note that context can override lexical preferences (see (30)). Given this fact, and the potential empirical problems even in "neutral" contexts (31), it is unclear just how important lexical preferences (as formulated by Ford et al.) are in the comprehension process. The issue might be clarified if lexical preferences could provide an explanation for garden path effects, but this proves quite problematic (see §2.4.4).

Weakly interactive models

Altmann, Crain, and Steedman, motivated by their own empirical work on the effects of context on parsing, have proposed a *fine-grained, parallel, weakly-interactive* sentence processing architecture (Crain & Steedman, 1985; Altmann & Steedman, 1988; Steedman & Altmann, 1989). A separate syntactic processor produces syntactic structures in parallel, on a word-by-word basis. The preferred analysis is selected on the basis of semantic or referential appropriateness. The model is *weakly* interactive because it maintains a separate

[9] Recall that the Sausage Machine could not predict RA effects on short sentences.

module that analyses the input syntactically without regard to semantics or context. It is *parallel* because, unlike the Minimal Attachment model, multiple structures are produced in parallel at local ambiguities. It is *fine-grained* because partial structures are produced and ambiguities resolved on a word-by-word basis.

The model particularly emphasizes the role of referential context in resolving ambiguities. This role is captured in two principles:

(39) *The Principle of Referential Support:* An NP analysis which is referentially supported will be favored over one that is not.

(40) *The Principle of Parsimony:* A reading which carries fewer unsupported presuppositions will be favored over one that carries more.

The operation of (39) can be illustrated in the following text:

(41) A psychologist was counseling two women. He was worried about one of the pair but not about the other. The psychologist told *the woman that*. . .

Referential Support will prefer attaching the incoming clause (signaled by *that*) to the NP *the woman* rather than as a complement of *told*, because the simple NP analysis of *the woman* fails to uniquely refer, since there are two women in the discourse context.

This model has much to recommend it, since it begins to explain how context can modulate the parsing process, without leaving these effects entirely to some unexplicated post-first-pass processing. The Principle of Parsimony also explains why certain apparently purely structural preferences obtain in neutral contexts (e.g., the preference for the main verb over reduced relative reading). What is not altogether clear in this approach, however, is why any modular effects of the kind listed in Table 2.3 should arise.

Weighted-evidence models

Weighted-evidence models refers to a class of processing theories that bring to bear multiple knowledge sources—syntactic, lexical, semantic, pragmatic—simultaneously and uniformly on each aspect of the comprehension process. In particular, syntactic parsing and the resolution of structural ambiguity are potentially under the influence of all these sources.

Such models may be called strongly interactive, or constraint-satisfaction models, but I have used the term weighted-evidence to emphasize a common feature. The support for alternative interpretations is some function of the different sources of evidence for that interpretation. These sources may have different weights or strengths. Processing decisions emerge as the result of a competition among representations of different interpretations.

Examples include the READER model (Thibadeau, Just, & Carpenter, 1982). READER is based on the CAPS architecture, which is an activation-based production system (Just & Carpenter, 1987). Productions (representing different knowledge sources) direct activation to representations of the input. The activation or strength of an interpretation is computed by summing the activation of all the productions supporting that interpretation. The amount of activation associated with each production is in turn a function of the strength of the particular production, and the activation level of representations satisfying the conditions of the production.

The connectionist model of (St. John & McClelland, 1990) implements similar ideas, though the network does not have distinguished bits of structure corresponding to productions. The St. John and McClelland model also differs from the READER model in an important and informative way: it does not compute a separate syntactic structure (cf. §2.1.3). The theory thus adopts a more radical form of nonmodularity. This is an independent choice which is not inherent to weighted-evidence or strongly interactive theories.

A central concern for these models as theories of ambiguity resolution is the difficulty in making specific predictions, since essentially any knowledge source may be posited to affect a choice. There is a need to continue the development of a knowledge level theory (along the lines of (39) and (40)) that will make predictions across a range of ambiguity situations with some generality.

Of course, another problem for strongly interactive theories is explaining why modular effects show up at all. The Capacity-Constrained READER model (CC READER) (Just & Carpenter, 1992) provides one interesting solution to this problem. In CC READER, modular effects arise because of limited working memory capacity, which corresponds to activation in the model. Activation limits may prevent all the relevant knowledge sources from being brought to bear. Furthermore, these limits are hypothesized to differ across individuals, predicting individual differences in modular effects. Although which knowledge sources suffer as a result of WM limits is still a degree of freedom in the model, the model does show in principle how an otherwise strongly interactive theory can exhibit modular effects.

Constrained parallelism

Parallel models of ambiguity resolution do not necessarily select one structure at an ambiguous point, but permit the explicit maintenance of multiple structures during the parse. The challenge in such a framework is to find ways to constrain the parallelism so that the number of interpretations does not grow combinatorially, and so that human performance is modeled.

The limit on activation in the CC READER model constrains its parallel structures (MacDonald, Just, & Carpenter, (1993)). If the limit is reached while maintaining multiple structures, one of the structures is discarded to free up resources. The model accounts for semantic and pragmatic preferences by piping activation to the more preferred structures, as described above. Again, while the basic structure is in place to handle a range of ambiguity resolution effects, one of the difficulties with the model is that there are many degrees of freedom in the strategies for working memory management, so it is difficult to make detailed predictions.

A series of more syntactically-oriented theories has been developed by Kurtzman (1985), Gorrell (1987), and Gibson (1991). These models rank parallel alternatives primarily according to structural features. As in the CC READER model, less preferred structures are continuously pruned. Gibson developed a precise metric for assigning values to structural alternatives, partly derived from principles of GB syntax. A structure is pruned if an existing structure is cheaper by some constant factor. Empirically, the model is an advance over other parallel theories of ambiguity resolution, because it is able to make detailed predictions and accounts for a range of structural preferences. The Gorell and Gibson models both allow for pragmatic and semantic effects to modulate the preferences, but this part of the theory is not worked out in much detail (in contrast to CC READER).

The theories above demonstrate that parallel models are capable in principle of exhibiting modular and interactive effects, and predicting structural preferences. But as mentioned in §2.3.4, parallel models do not automatically account for the complex phenomena surrounding parallelism in humans.

Deterministic parsers

The PARSIFAL system of Marcus (1980) was the first attempt to build a model of parsing that was strictly deterministic. Marcus proposed the following definition of a deterministic parser:

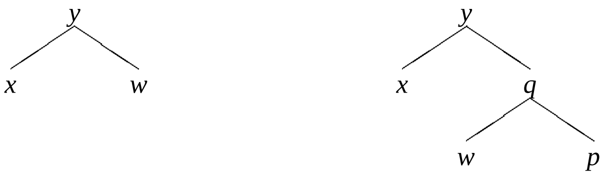
1. All syntactic substructures created by the parser are permanent.
2. All syntactic substructures created by the parser for a given input must be output as part of the structure assigned to that input.
3. The internal state of the parsers must be constrained in such a way that temporary syntactic structures are not encoded within the state of the machine (no simulated non-determinism).

Strictly deterministic parsers are simpler and more efficient than nondeterministic parsers since they dispense with the additional mechanisms required to maintain parallel states or perform backtracking.

Marcus noted that a deterministic parser that is forced to make choices immediately at ambiguous points falls short of the apparent human capacity to handle ambiguities, overpredicting garden path effects. For this reason, PARSIFAL uses lookahead to delay attachment decisions. The lookahead is supported by a three cell buffer that holds constituents awaiting attachment. Attachment decisions are made by pattern-action rules sensitive to the future syntactic context.

The relative simplicity and efficiency of deterministic architectures makes them appealing candidates for psycholinguistic models. However, as a model of ambiguity resolution, PARSIFAL has two fairly serious shortcomings. First, the systematic delay of attachment decisions is inconsistent with the accumulated evidence in support of syntactic immediacy (§2.2.1). Second, the strategies (rules) used to resolve ambiguities are purely syntactic, and therefore subject to the same criticism of all syntactic resolution theories: the inability to account for the interactive effects documented in §2.3.3. In principle, however, it should be possible to construct deterministic lookahead models that are at least weakly interactive.

Subsequent work by Marcus et al. (1983) led to a variant of deterministic parser known as *minimal commitment* parsers (Weinberg, 1993; Gorrell, 1993). Minimal commitment



(a) (b)

D(y,x), D(y,w)

D(y,x), D(y,w), D(y,q), D(q,w), D(q,p)

FIGURE 2.2: Dominance relations in minimal commitment models.

parsers dispense with lookahead and instead adopt a representation of syntactic structure known as *D-theory*, which permits underspecification. The key is building trees with dominance relations that do not encode strict dominance. In this way, the parser can minimally commit to structure. Figure 2.2 shows how this works. The representation of the tree in (a) can be changed to a representation of the tree in (b) by simply adding a new dominance relation. The original relation $D(y,w)$ does not assert that *y directly* dominates *w*, so any number of phrasal nodes can be added later between *y* and *w*, as long as *y* continues to dominate *w*.

Since minimal commitment theories do not use lookahead, they offer the possibility of addressing immediacy of interpretation. Weinberg (1991) assumes that the representations are, in fact, immediately interpreted. But this requires determining which syntactic relations actually hold. Because the dominance relations do not explicitly specify which relations hold, additional computation must be performed to make the immediate dominance relations explicit[10]. These computations are purely syntactic in nature, since they process and produce purely syntactic representations. Whether this process is considered part of the parser itself, or assigned to the semantic interpreter—an odd partioning of function—determinism is violated since the structures produced for interpretation may change nonmonotonically throughout the parse. Thus, the minimal commitment models still have not reconciled determinism with immediacy. Furthermore, these models, like their predecessors, adopt purely syntactic resolution strategies, leading to the problems discussed above.

[10] In general, this may require computing the transitive closure of the dominance relations. Let $I(x,y)$ mean that x immediately dominates y. (These are the relations required for interpretation.) The dominance relations can be taken as assertions in the predicate calculus. Then the following axioms will suffice to compute immediacy relations:

$$\forall x,y\,[D(x,y) \land \neg\exists w\,(D(x,w) \land D(w,y)) \rightarrow I(x,y)]$$

$$\forall x,y,z\,[D(x,y) \land D(y,z) \rightarrow D(x,z)]$$

2.4 Garden path effects and unproblematic ambiguities

In this section we review the range of garden path (GP) effects and unproblematic ambiguities (UPA). The section concludes with a review of previous garden path theories. Though GP/UPA phenomena are so closely related to ambiguity resolution (§2.3) that they might properly be considered subphenomena, I treat them separately for two reasons. First, the richness of the phenomena deserves focused attention of its own. Second, a theory of ambiguity resolution is not automatically a theory of garden path effects. In other words, it is possible to have a good theory of ambiguity resolution without having a good theory of GP/UPA effects, and vice versa. This should become clear in §2.4.4.

2.4.1 Garden path phenomena defined

A garden path effect arises when a reader or listener misinterprets a local ambiguity in some way and is unable to recover the correct interpretation without reprocessing. The result is an impression that the sentence is ungrammatical or does not make sense. There are two kinds of garden path effects, as shown in (42) and (43):

(42) The cotton clothing is made of grows in Mississippi. (cf. *The cotton that clothing is made of grows in Mississippi.*)

(43) (a) War Worries Dog Consumers (cf. *War worries are dogging consumers*)[11]
(b) The old man's glasses were filled with sherry.

Syntactic garden paths, such as (42), arise because the disambiguating information is grammatical in nature, and the structure assigned to the ambiguous material cannot be grammatically incorporated into the remainder of the sentence. Syntactic GPs give rise to an impression of ungrammaticality. *Nonsyntactic garden paths*, such as those in (43), arise because the disambiguating information is semantic or pragmatic in nature. The interpretation and structure assigned to the ambiguous material can be grammatically incorporated into the remainder of the sentence, but not without creating a semantic or pragmatic anomaly—often to humorous effect. There are several kinds of nonsyntactic garden paths, depending on whether the initial ambiguity is structural (43a) or semantic (43b). I shall have little more to say about nonsyntactic GPs since they have not been well-studied. For the remainder of the thesis, the term *garden path* refers to syntactic garden path effects unless otherwise noted.

The definition of GP given above is not universally adopted in the psycholinguistic literature, though it is fairly common. Another frequent use of the term is to refer to any measurable effect that results from subjects making a wrong choice at a local ambiguity, regardless of whether the misinterpretation give rise to impressions of ungrammaticality. While those effects are certainly interesting as well, I continue to use the stronger definition because it emphasizes an empirically measurable and theoretically important distinction.

[11] Discovered by *The New Yorker*, date unknown.

The role of garden path effects in psycholinguistics

GP effects have played a dual role in psycholinguistics. Most commonly, they have been used as a diagnostic for exploring strategies of ambiguity resolution, because a GP effect is strong evidence that a subject chose a particular path at an ambiguous point. However, the GP effects themselves have seldom been systematically explored. The result has been a substantial body of ambiguity resolution theory (§2.3) with few truly adequate accounts of GP effects per se. This general point has also been emphasized by Pritchett (1992).

Evidence for garden path effects

Despite the fact that GP effects have been somewhat neglected in the field, there exists a fair amount of compelling evidence for a range of GP types. There are two reasons for this: the perceived ungrammaticality of GPs makes them easily subject to linguistic intuition, and the raft of experiments on ambiguity resolution have left behind a gold mine of data relevant to GPs.

Tables 2.5, 2.6, and 2.7 presents a collection of GP effects classified by syntactic type. (The syntactic classification is meant to be descriptive; the use of traditional grammatical relations should not be taken as a theoretical commitment to a relational or functional grammar). The strongest source of empirical evidence comes from *rapid grammaticality judgments*, where subjects indicate whether they think a sentence or sentence fragment is grammatical within a few hundred milliseconds of reading it. Reading times provide additional evidence, but must be interpreted with care, since reading times alone do not always distinguish GP from non-GP cases. No GP type is listed solely on the basis of reading times.

When examples of these garden path types are presented in the text, the GP number from Tables 2.5–2.7 will be appended to the example number as follows:

(44; GP1) The businessman sent the paintings yesterday was upset.

2.4.2 Unproblematic ambiguities

If the story ended with GP effects, there would be no difficult theoretical problem: any single-path deterministic parser would make the right predictions. However, Marcus (1980) pointed out that there are ambiguous structures which do not cause people difficulty no matter which interpretation of the ambiguity proves correct. Tables 2.8, 2.9, and 2.10 list a range of such unproblematic ambiguities. UPA examples come in pairs. Each sentence in the pair requires the local ambiguity to be interpreted in a different way, but both sentences are acceptable. All the types involve structural ambiguity of some kind, with the exception of UPA30 and UPA31, which were included to illustrate the insufficiency of semantic (particularly thematic role) ambiguity to cause GP effects.

TABLE 2.5: A collection of garden path constructions (part 1 of 3).

	TYPE	EXAMPLE
GP1	Direct object/subject (Frazier & Rayner, 1982; Pritchett, 1988)	Since Jay always jogs a mile seems like a short distance to him. (cf. *Since Jay always jogs, a mile seems like a short distance to him.*)
GP2	Direct object/subject (long) (Warner & Glass, 1987)	The girls believe the man who believes the very strong ugly boys struck the dog killed the cats. (cf. *The girls believe that the man who believes that the very strong ugly boys struck the dog killed the cats.*)
GP3	Complement clause/subject sentence (Gibson, 1991)	I believe that John smokes annoys Mary. (cf. *I believe that John's smoking annoys Mary.*)
GP4	Direct object/subject with embedded relative (Warner & Glass, 1987)	Before the boy kills the man the dog bites strikes. (cf. *Before the boy kills, the man that the dog bites strikes.*)
GP5	Direct object/subject with relative clause (Warner & Glass, 1987)	When the horse kicks the boy the dog bites the man. (cf. *When the horse kicks the boy, the dog bites the man.*)
GP6	Preposition object/subject (Frazier, 1978; Pritchett, 1988)	Without her contributions failed to come in. (cf. *Without her, contributions failed to come in.*)
GP7	Direct object/subject of second complement (Pritchett, 1988)	I convinced her professors hate me. (cf. *I convinced her that professors hate me.*)
GP8	Direct object/subject of second complement (optional first complement) (Pritchett, 1988)	The doctor warned the patient would be contagious. (cf. *The doctor warned that the patient would be contagious.*)

TABLE 2.6: A collection of garden path constructions (part 2 of 3).

	TYPE	EXAMPLE
GP9	Indirect object/subject of relative (Wanner et al., 1975; Pritchett, 1992)	John gave the boy the dog bit a dollar. (cf. *John gave the boy that the dog bit a dollar.*)
GP10	Embedded object/matrix object (Pritchett, 1992)	Sue gave the man who was racing the car. (cf. *Sue gave the car to the man who was racing.*)
GP11	Complement clause/relative clause (Crain & Steedman, 1985)	The psychologist told the wife that he was having trouble with to leave. (cf. *The psychologist told the wife who(m) he was having trouble with to leave.*)
GP12	PP argument/adjunct (Gibson, 1991; Pritchett, 1992)	I sent the letters to Ron to Rex. (cf. *I sent the letters from Ron to Rex.*)
GP13	Relative clause/complement clause (Crain & Steedman, 1985)	The psychologist told the wife that he was having trouble with her husband. (cf. *The psychologist told the wife that he was having trouble with to leave.*)
GP14	Main verb/reduced relative (Bever, 1970)	The horse raced past the barn fell. (cf. *The car driven past the barn stalled.*)
GP15	Main verb/reduced relative (short) (Kurtzman, 1985; Abney, 1989)	The boat floated sank. (cf. *The car driven stalled.*)
GP16	Ditransitive main verb/reduced relative (Rayner et al., 1983)	The woman brought the flowers smiled broadly. (cf. *The woman given the flowers smiled broadly.*)
GP17	Main verb/embedded relative (Gibson, 1991)	The dog that was fed next to the cat walked to the park chewed the bone. (cf. *The dog that was fed next to the cat seen by the boy chewed the bone.*)

TABLE 2.7: A collection of garden path constructions (part 3 of 3).

	TYPE	EXAMPLE
GP18	Adjective/noun followed by noun/verb (Milne, 1982)	The building blocks the sun faded are red. (cf. *The blocks that the sun faded are red.*)
GP19	Noun/verb (Milne, 1982)	The granite rocks by the seashore with the waves. (cf. *The granite gently rocks by the seashore with the waves.*)
GP20	Adjective-noun/noun-relative (Marcus, 1980)	The cotton clothing is made of grows in Mississippi. (cf. *The cotton that clothing is made of grows in Mississippi.*)
GP21	Subject/verb (derived nominal) (Pritchett, 1992; Milne, 1982)	The old train the young. (cf. *The older folks train the younger folks.*)
GP22	Predicate complement/subject (Ford et al., 1982)	The boy got fat melted. (cf. *The boy got butter melted.*)
GP23	*That* complementizer/pronoun	Before she knew that she went to the store. (cf. *Before she knew that, she went to the store.*)
GP24	*That* complementizer/determiner (Gibson, 1991)	I saw that white moose are ugly. (cf. *I saw that cats are ugly.*)
GP25	*That* complementizer for subject sentence/determiner (Gibson, 1991)	That coffee tastes terrible surprised John. (cf. *It surprised John that coffee tastes terrible.*)
GP26	Main verb/auxiliary (Kurtzman, 1985; Marcus, 1980)	Have the boys given gifts by their friends. (cf. *Have the boy's friends give gifts to them.*)

TABLE 2.8: A collection of unproblematic ambiguities (part 1 of 3).

	TYPE	EXAMPLE
UPA1	Direct object/subject (Kimball, 1973; Ferreira & Henderson, 1990)	I knew the man. I knew the man hated me passionately.
UPA2	Direct object of fronted clause/subject (short) (Warner & Glass, 1987)	When the boys strike the dog kills. When the boys strike the dog the cat runs away.
UPA3	Direct object/subject (long) (Pritchett, 1992)	Ron believed the ugly little linguistics professor. Ron believed the ugly little linguistics professor he had met the week before in Prague disliked him.
UPA4	NP/NP specifier (Pritchett, 1988)	Without her we failed. Without her contributions we failed.
UPA5	Plural NP/NP specifier (Pritchett, 1988)	The woman kicked her sons. The woman kicked her sons dogs houses doors.
UPA6	Second object/specifier (Gibson, 1991)	The cop gave her earrings. The cop gave her earrings to the dog.
UPA7	PP argument/argument (Gibson, 1991)	The minister warned the president of the danger. The minister warned the president of the republic of the danger.
UPA8	Predicate complement/NP-modifier (Marcus, 1980)	Is the block in the box? Is the block in the box red?
UPA9	Complement/subject relative (Gibson, 1991)	John told the man that Mary kissed Bill. John told the man that kissed Mary that Bill saw Phil.
UPA10	NP complement/relative clause (Gibson, 1991)	The report that the president sent to us helped us make the decision. The report that the president sent the troops into combat depressed me.

TABLE 2.9: A collection of unproblematic ambiguities (part 2 of 3).

	TYPE	EXAMPLE
UPA11	Predicate complement/adjective	The boy got fat. The boy got fat mice for his pet snake.
UPA12	Main verb/reduced relative, obligatory object (Pritchett, 1988; Ferreira & Clifton, 1986; Just & Carpenter, 1992)	The defendant examined the evidence. The defendant examined by the lawyer shocked the jury.
UPA13	Reduced relative/main verb (Gibson, 1991; Pritchett, 1992)	The bird found in the room died. The bird found in the room enough debris to build a nest.
UPA14	Modified main verb/reduced relative	The defendant carefully examined the evidence. The defendant carefully examined by the prosecutor looked nervous.
UPA15	Compound noun followed by noun/verb (Frazier & Rayner, 1987)	The warehouse fires numerous employees each year. The warehouse fires kill numerous employees each year.
UPA16	Noun/auxiliary verb (Gibson, 1991)	The paint can fell down the stairs. The paint can be applied easily with a new brush.
UPA17	Adjective/noun followed by noun/verb (Milne, 1982)	The building blocks are red. The building blocks the sun.
UPA18	Noun/adjective	The square is red. The square table is red.
UPA19	Complement/adjective (Pritchett, 1988)	I like green. I like green dragons.
UPA20	Derived nominal (Milne, 1982; Pritchett, 1992)	The old teach very well. The old train is big.

TABLE 2.10: A collection of unproblematic ambiguities (part 3 of 3).

	TYPE	EXAMPLE
UPA21	*That* pronoun/determiner	I know that. I know that boy.
UPA22	*That* pronoun/complementizer	I know that. I know that dogs should play.
UPA23	Singular noun/plural noun (Kurtzman, 1985)	The sheep seem very happy. The sheep seems very happy.
UPA24	*To* inflection marker/preposition (Gibson, 1991)	I opened the letter to Mary. I opened the letter to impress Mary.
UPA25	Object gap/preposition object gap	John saw the ball the boy hit. John saw the ball the boy hit the window with.
UPA26	Long distance object gap	Who do you believe? Who do you believe John suspects Steve knows Bill hates?
UPA27	NP/small clause VP (Pritchett, 1992)	I saw her duck fly away. I saw her duck into an alleyway.
UPA28	Coordination	I went to the mall. I went to the mall and the bookstore.
UPA29	Multiple compounding (Pritchett, 1992)	We admire their intelligence. We admire their intelligence agency policy decisions.
UPA30	Semantic role switch (Pritchett, 1992; Tanenhaus & Carlson, 1989)	I gave the dogs to Mary. I gave the dogs some bones.
UPA31	Verb/verb plus particle	John picked the boy for his team. John picked the boy up yesterday.

2.4.3 Some general garden path phenomena

In addition to simply compiling the known garden path constructions, we can abstract a few general facts about GP/UPA phenomena, most of which are apparent from examination of Tables 2.5–2.10.

1. *Recoverability.* People can eventually recover from garden paths through deliberation or explicit instruction. Once the "puzzle" is solved, the sentence may be perceived as grammatical. Recoverability is theoretically significant in itself and also helps to distinguish GPs from other kind of processing difficulties, such as those discussed in §2.5.

2. *Bidirectionality.* GP effects can be bidirectional in the sense that they are independent of any preferred direction of the resolution of an ambiguity. In other words, GP effects can arise even when the unpreferred path is taken at a local ambiguity (say, the relative clause reading over the main verb reading) and the normally preferred interpretation turns out to be correct. Examples include GP4 and GP13. This clearly demonstrates the independence of the GP effect from the phenomena of ambiguity resolution per se.

3. *Independence of length.* Length is not necessarily a determining factor in GP effects (Pritchett, 1992). More precisely, the distance from the ambiguous point to the disambiguating region may be very short, or even zero, and still give rise to a GP effect (GP1, GP6–10, GP15, GP22); and the distance to the disambiguating region may be extended without necessarily giving rise to a GP effect (UPA3,UPA8).

4. *Distance-to-disambiguation effects.* Although length is not always a factor, the material intervening between the initial ambiguity and the disambiguating point *can* have an effect on both the immediate perception of grammaticality (GP2; (Warner & Glass, 1987)), and the process of deliberate recovery (Ferreira & Henderson, 1991). Generally, these studies found that the more intervening material, the more likely a GP effect arises or the more difficult it is to recover from. Warner & Glass (1987) claim that this is essentially a length effect, but length alone (as measured in some surface metric such as words or syllables) cannot be the sole factor as demonstrated above. Therefore, "distance-to-disambiguation" refers to the weaker claim that the intervening material can have an effect; that much is supported by the data.

5. *Independence of lexical ambiguity*. Lexical ambiguity is neither necessary nor sufficient for GP effects to arise Pritchett (1992). This is apparent from GP1–17, UPA15–24, and UPA27.

6. *Independence of semantic content.* Semantic ambiguity need not cause a GP effect. UPA30 exhibits local ambiguity in the assignment of thematic roles, but both sentences are easily processed (Tanenhaus & Carlson, 1989; Pritchett, 1992):

   (45; UPA30) (a) We loaded the truck$_{GOAL}$ with bananas$_{THEME}$
   (b) We loaded the truck$_{THEME}$ onto the ship$_{GOAL}$.

2.4.4 Garden path theories

"Garden Path" models

Garden Path[12] models (Clark & Clark, 1977) are simply single-path models—they maintain a single interpretation during comprehension. GP effects thus arise whenever the comprehender selects the wrong interpretation at an ambiguity.

The UPA data should make unabundantly clear that any purely single-path theory is doomed to overpredict GP effects, no matter how accurately that theory may predict the direction of ambiguity resolution. Nevertheless, the Garden Path model is the default assumption in many theories, such as the early strategy-based models[13], the Sausage Machine/Minimal Attachment model, and the semantically-driven model of Milne (1982).

To make the point concrete, consider the failure of Minimal Attachment on (46) below:

(46; UPA1) (a) Seth believed the director.
(b) Seth believed the director was lying.

Minimal Attachment predicts that the NP *director* will be attached in complement position, as in (46a). Therefore, when this proves to be incorrect in (46b), a garden path effect should arise; this is clearly not the case. Similar criticisms hold of the Altmann/Crain/Steedman model, strictly interpreted as a Garden Path model (Gibson, 1991).

Deterministic parsers

The PARSIFAL model (Marcus, 1980) fares somewhat better than the Garden Path theories described above, since the basic lookahead architecture is fundamentally responsive to the need to handle unproblematic ambiguities. The lookahead buffer in PARSIFAL holds constituents[14] rather than words (as in the six-word window of the Sausage Machine). Garden path effects arise when the disambiguating syntactic material falls outside the three-cell window, and the parser operates short-sightedly. For example, consider processing on (47a) below:

(47; GP14) (a) The boat floated down the river sank.
(b) The boat [floated]$_V$ [down]$_P$ [the river]$_{NP}$.

After processing the NP *the boat* (which is then pushed onto the stack), the contents of the lookahead cells are as shown in (47b). The disambiguating final verb *sank* is out of sight as

[12] Just to be clear: *Garden Path* will be used to refer to this particular class of theories, while *garden path* refers to the phenomenon. *GP* is an abbreviation of *garden path*. Thus, Garden Path models are a class of garden path (GP) theory. The *Garden Path theory* is sometimes used to refer specifically to Frazier's Minimal Attachment theory (Frazier, 1987), but not in this thesis.

[13] Though Kimball (1973) realized the problem and assumed human comprehension employed some lookahead.

[14] The precise specification of which constituents occupied cells in the buffers was apparently a degree of freedom in the model for fitting GP data; see (Pritchett, 1992) for discussion.

the processor structures the initial material. The main verb reading is chosen, and the GP effect subsequently ensues when *sank* cannot be incorporated.

The primary empirical problem with PARSIFAL's lookahead device (apart from the inconsistency with immediacy discussed in §2.3.5) is that it predicts length effects where none exist. Consider just a slight variant on the GP structure in (47a):

(48; GP14) The boat floated quickly sank.

Since the remainder of the sentence after *the boat* can fit into the three-cell buffer, PARSIFAL incorrectly predicts that this sentence is not a garden path.

The later D-theory model developed by Marcus et al. (1983) traded the power of the lookahead buffer to resolve ambiguities for the power of dominance relations to minimally commit to syntactic structure. The D-theory model is an advance insofar as it eliminates the oversensitivity to length. Unfortunately, the move to D-theory also trades one set of empirical problems for another. In particular, D-theory encounters difficulty with GP7 and GP8; see (Pritchett, 1992) for details.

The minimal commitment models of Weinberg (1993) and Gorrell (1993) modify various aspects of the original D-theory model, improving upon its predictions. As noted in §2.3.5, however, two problems remain for all the theories developed thus far within the deterministic framework: accounting for immediacy of interpretation, and accounting for the interactive ambiguity resolution effects.

Garden Path models with constrained reanalysis

To adequately account for the UPA data, a single-path model must be augmented with some kind of reanalysis mechanism. The reanalysis must be constrained in some fashion, otherwise the GP predictions would be lost. Frazier & Rayner (1982) proposed a kind of reanalysis to augment the Minimal Attachment model. The motivation was precisely the kind of unproblematic structure exhibited in (46). The reanalysis strategy (dubbed *Steal-NP* by Pritchett (1992), after (Abney, 1986)) is triggered by an incoming subjectless verb, for example, *was* in (46b). A previously analysed NP (*the director*) can then be attached as the subject of the incoming verb, provided the NP is close enough in the surface string.

Steal-NP was an advance over other Garden Path models because it was the first explicit attempt to formulate a reanalysis strategy. However, there are a number of shortcomings. Apart from the problems of vague formulation, the strategy fails to actually make the correct predictions regarding GP and UPA contrasts (Pritchett (1992) presents a detailed critique). For example, while the ease of processing (46) is accounted for, the GP predictions for the following structures are now missed:

(49; GP1) While Mary sewed a sock fell on the floor.

(50; GP8) Sharyn warned the professor would be angry.

Blank (1989) proposes a much more computationally explicit reanalysis mechanism called *boundary backtracking.* Blank's parsing architecture is based on a grammatical formalism called register vector grammar. The architecture is basically a finite state machine, where state symbols are replaced with vectors of three-valued syntactic features (the three values are +, -, and don't care). Parsing is accomplished by making transitions through the state space with productions that match against the state vector and make changes to the vector. There exists a fixed number of boundary registers that save the state of the machine at particular phrase boundaries types. If the parse fails, the machine returns to a state in one of the boundary registers to try a different path. If the required state is not available, the machine fails.

As an example of how Blank's machine predicts a garden path, consider processing on the main verb/reduced relative GP:

(51; GP14) The horse | raced past | the barn | fell.

Relevant boundaries are marked with a | in the sentence. Blank posits a boundary register that saves state at phrasal boundaries. As the processor encounters *raced,* an explicit production fires triggered by the closing of the noun phrase *horse.* This production saves the current state in the phrasal boundary register. This is the state just before the noun phrase is closed, and includes the option of taking *raced* as a reduced relative modifier. After the processor accepts the preposition *past*, it again saves state in the phrasal boundary register, overwriting the old state. Thus, when the disambiguating final verb occurs, and the processor attempts to backtrack, the required state is no longer available.

One theoretical problem, which Blank himself points out, is that the type and number of the boundary registers is unconstrained—Blank simply chose a set that seemed reasonable. There are a number of empirical problems as well, since the model is oversensitive to length effects (e.g., GP15). However, it remains an important contribution since it is the first computational system not based on lookahead that makes explicit GP/UPA predictions.

On-line Locality Constraint

As part of his program of developing a strongly grammatically-derived processing model, Pritchett (1988, 1992) proposed a constraint that characterizes precisely when garden path effects arise, as a function of the structure of the preferred interpretation and the structure of the required (globally correct) interpretation:

> (52) *The On-line Locality Constraint (OLLC):* The target position (if any) must be *governed* or *dominated* by the source position (if any), otherwise attachment is impossible by the automatic Human Sentence Processor.

We need not be concerned here with the precise definitions of government and dominance—only that they are grammatical relationships central to GB syntax, and that they are purely structural relationships defined between nodes of phrase structure trees.

To see how the OLLC predicts GP effects, consider the garden path in (53):

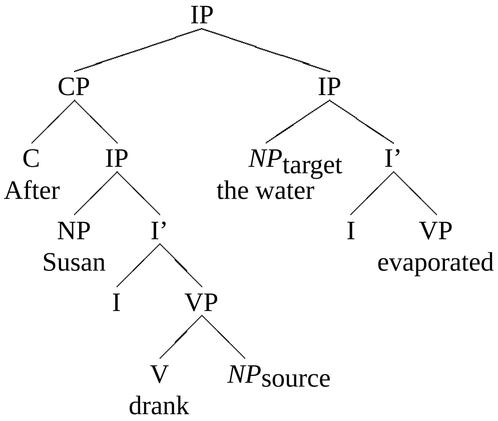


FIGURE 2.3: How the On-line Locality Constraint (OLLC) predicts a garden path. The NP *the water* is initially attached as the complement of the VP *drank* (source position). The globally correct position (target) is not governed or dominated by the source position, violating the OLLC.

(53; GP1) After Susan drank the water evaporated.

The globally correct structure of (53) is given in Figure 2.3. Assume that *the water* is initially attached as the object of *drank* (as is required by Generalized Theta Attachment (36)). The source and target positions of the NP *the water* are annotated on the tree structure. The relevant fact is that *the water* must be reanalysed from the object of *drank* (source) to the subject of *evaporated* (target). (In GB terms, from the complement of a VP to the specifier of an IP). However, as should be apparent from Figure 2.3, the source position neither dominates nor governs the target position, in violation of OLLC.

The OLLC (and the original Theta Reanalysis Constraint that it replaces) is an important breakthrough in GP theory for three reasons. First, it provides a precise and widely applicable reanalysis constraint: the OLLC can be applied to *any* structure in question, provided a GB analysis can be provided. Second, at the time, the theory provided by far the widest empirical coverage with respect to the range of garden paths and unproblematic ambiguities. Third, the work established support for the more general claim that GP effects are purely a function of syntactic structure.

The latter point is so important that I present it as a separate hypothesis below:

(54) *The Structural Garden Path Hypothesis:* Garden path effects are purely a function of differences between the syntactic structure of the preferred interpretation, and the syntactic structure of the globally correct interpretation.

This is a more general claim than the OLLC; the OLLC is one possible way of defining the relevant difference between structures. (54) does not specify what the preferred interpretation is—in particular, it is *not* saying that the preferred interpretation is purely a function of syntactic structure. That is an altogether stronger and independent position, one that Pritchett in fact adopts in the form of Generalized Theta Attachment (36).

Frequency/strength based accounts

A widely-adopted assumption is that GP effects are a function of the strength or frequency of particular lexical or syntactic forms. Under this view, GP effects arise when a stronger or more frequent lexical/syntactic form must be abandoned in favor of a form that is much weaker or less frequent. Ford et al. (1982) formulated a lexical version of this hypothesis which attributed GP effects to morphosyntactic reanalysis of lexical items. Consider their example:

(55; GP22) The boy got fat melted.

According to the lexical theory, the GP effect in (55) arises because the strong lexical form for *fat* is adjectival, and this analysis must be dropped in favor of a nominal analysis. A similar explanation holds of the familiar main verb/reduced relative ambiguities GP14: the active form of the verb is much stronger than the passive participle. The theory runs into empirical difficulty by overpredicting GP effects. For example, neither sentence in (56) causes difficulty:

(56; UPA11) (a) Mike likes fat.
(b) Mike likes fat steaks.

In general, a purely lexical GP theory cannot account for the fact that lexical ambiguity is neither sufficient nor necessary for garden path effects to arise (§2.4.3).

A similar explanation for garden path effects is often given with respect to syntactic structures rather than lexical forms. For example, the CC READER model (Just & Carpenter, 1992) and Jurafsky (1992) model assume the main verb/reduced relative garden path derives from the much higher frequency or strength of the matrix clause structure over the reduced relative structure (in effect, evoking the Canonical Sentoid Strategy). There are three potential problems with such explanations; the first is methodological, the second and third empirical:

1. It is difficult to make any predictions with the theory, without actually obtaining some frequency counts of syntactic structures in naturally occurring texts. Furthermore, even if such counts could be obtained and found to be consistent with some GP effects, it is possible the underlying cause of garden paths could still be missed. Reduced relative structures may be less frequent in part *because* they lead to garden paths, not the other way around. Schlesinger (1968) notes this problem with frequency-based theories of comprehension difficulty.

2. It seems unlikely that construction frequencies will account for the range of effects in Tables 2.5–2.10. For example, it is doubtful that there are any significant frequency differences between complement clause and relative clause constructions (GP11).

3. Frequency/strength theories may have difficulty accounting for the *bidirectionality* of GP effects noted in §2.4.3. These phenomena demonstrates that garden paths are independent of whatever is considered to be the strongest or most preferred structure.

A further challenge for these theories is presented by the comparatively overwhelming success of structurally-based models which have no role for relative frequencies or strengths (e.g., the On-line Locality Constraint discussed above).

Constrained parallel models

Constrained parallel models (or *limited path* models) derive their GP/UPA predictions from constraints that limit the structural interpretations that may be carried along in parallel. If the disambiguating material is reached and the required structure is still available, no garden path effect arises. If the disambiguating material is reached and the required structure has been pruned for some reason, a garden path effect does arise and the parse fails. The constraints that limit parallel structures play the same theoretical role that reanalysis constraints play in single path models.

CC READER (Just & Carpenter, 1992) is a paradigmatic example of a constrained parallel model. In CC READER, the fixed amount of available activation in the system limits the structures that may be maintained in parallel. The model embodies directly the hypothesis that GP effects emerge because of working memory limitations. To actually derive predictions from the model requires specifying how structures will differentially consume the activation resource and how the system responds to potential overflows. While the model could in principle account for a range of effects, these necessary specifications have not been worked out in detail. To the extent that they are specified, the predictions depend on different frequencies or strengths of alternative structures, leading to the difficulties described in the section above.

The parallel model in (Gibson, 1991) does present a detailed set of structural metrics and principles for pruning interpretations, derived primarily from GB syntax. These principles, in effect, are another instantiation of the Structural Garden Path Hypothesis (54). While presenting an example of the theory at work would require introducing too much detail here, the important fact to note is that this model shares many of the strengths of the On-line Locality Constraint presented earlier: the theory is applicable to any structure that can be given a GB analysis, and it accounts for a wide range of GP/UPA effects. It therefore clearly establishes that detailed accounts of garden path effects may be developed within the constrained parallel framework. Although loosely motivated by working memory capacity constraints, the role of working memory is not as clear in this model as in CC READER.

2.5 Parsing breakdown and acceptable embeddings

Center-embedded (or self-embedded) sentences such as (57) were among the first constructions studied by psychologists and linguists concerned with the distinction between linguistic competence and performance (Miller & Chomsky, 1963; Miller & Isard, 1964).

(57) The cat that the bird that the mouse chased scared ran away.

Center-embedded constructions are interesting because they are notoriously difficult to comprehend, but cannot be ruled out as ungrammatical without introducing ad hoc restrictions into the grammar to prevent embeddings beyond a certain level. They are the paradigmatic example of an *unacceptable* but *grammatical* sentence (Chomsky, 1965). Thus, most psychologists and linguistics alike have assumed that there must be a psychological rather than linguistic explanation of the difficulty with center-embeddings (e.g., (Miller & Chomsky, 1963; De Roeck et al., 1982)).

2.5.1 Parsing breakdown defined

The unacceptability of (57) is an example of the general phenomenon of *parsing breakdown* (PB). Parsing breakdown occurs when a listener or reader is unable to comprehend and perceive a sentence as grammatical without great difficulty. In this broad sense, parsing breakdown includes garden path effects, but I will generally use the term to refer only to breakdown that cannot be attributed to misinterpreting local ambiguities. This section is concerned with just this narrower class of breakdown effects.

Evidence for parsing breakdown

Nearly all of the experimental evidence bearing on parsing breakdown involves the center-embedded construction introduced above. The basic finding is that doubly-embedded object relative clauses, as in (57), cause great difficulty. This has been demonstrated in a number of ways: subjects consistently judge center-embeddings ungrammatical (Blumenthal, 1966; Marks, 1968), and perform poorly on simple verbatim recall tasks (Miller & Isard, 1964; Foss & Cairns, 1970), untimed paraphrase tasks (Blumenthal, 1966; Stolz, 1967; Larkin & Burns, 1977), and questions that test comprehension (Blauberg & Braine, 1974). In most of these experiments, the baseline for comparison was performance on right-branching sentences such as (58), which carry the same amount of content in a different syntactic structure.

(58) The bird chased the mouse that scared the cat that ran away.

Performance on right-branching versions of center-embedded sentences was always better and did not show the severe decrement at two levels of embeddings that the center-embedded sentences did.

Because parsing breakdown (like garden path effects) can be revealed by linguistic acceptability judgments, linguists and other researchers have generated a wide range of unacceptable but (putatively) grammatical structures. Tables 2.11 and 2.12 present a list derived primarily from Gibson (1991), which itself drew heavily on Cowper (1976). Although the unacceptability of most of these sentences was determined by informal survey and therefore not subject to the rigors of multiple experiments and multiple experimental paradigms, it nevertheless convincingly demonstrates that the phenomenon of parsing breakdown extends beyond just object-relative center-embeddings—just as the garden path effect extends beyond the canonical main verb/reduced relative construction.

TABLE 2.11: A collection of constructions causing parsing breakdown (part 1 of 2).

	TYPE	EXAMPLE
PB1	Center-embedded object-relative (Miller & Chomsky, 1963; Miller & Isard, 1964)	The man that the woman that the dog bit likes eats fish.
PB2	Center-embedded object-relative, dropped complementizers	The man the woman the dog bit likes eats fish.
PB3	Center-embedded subject-relative in Wh-question (Gibson, 1991)	Who did John donate the furniture that the repairman that the dog bit found to?
PB4	Center-embedded subject-relative (Gibson, 1991)	The man that the woman that won the race likes eats fish.
PB5	Embedded subject sentence (Kimball, 1973)	That that Joe left bothered Susan surprised Max.
PB6	Relative clause with embedded subject sentence (Gibson, 1991)	The woman that for John to smoke would annoy works in this office.
PB7	Post-verbal relative clause with embedded subject sentence (Gibson, 1991)	The company hired the woman that for John to smoke would annoy.
PB8	Subject sentence embedded in sentential complement (Gibson, 1991)	Mary's belief that for John to smoke would be annoying is apparent due to her expression.
PB9	Embedded sentential complement (Gibson, 1991)	John's suspicion that a rumor that the election had not been run fairly was true motivated him to investigate further.

TABLE 2.12: A collection of constructions causing parsing breakdown (part 2 of 2).

	TYPE	EXAMPLE
PB10	Sentential complement embedded in relative clause (Gibson, 1991)	The man who the possibility that students are dangerous frightens is nice.
PB11	Wh-question with sentential complement with embedded relative (Gibson, 1991)	Who does the information that the weapons that the government built don't work properly affect most?
PB12	Cleft with modified sentential complement (Gibson, 1991)	It is the enemy's defense strategy that the information that the weapons that the government built didn't work properly affected.
PB13	Clefted subject sentence (Gibson, 1991)	It is the enemy's strategy that for the weapons to work would affect.
PB14	Pseudo-cleft with modified sentential complement (Gibson, 1991)	What the information that the weapons that the government built didn't work properly affected was the enemy's defense strategy.
PB15	Pseudo-cleft with subject sentence (Gibson, 1991)	What for the weapons to work properly would affect is the enemy's defense strategy.
PB16	Though-preposing with modified sentential complement (Gibson, 1991)	Surprising though the information that the weapons that the government built didn't work properly was, no one took advantage of the mistakes.
PB17	Though-preposing with subject sentence (Gibson, 1991)	Surprising though for the weapons to work properly would be for the general populace, it would not surprise some military officials.

2.5.2 Acceptable embeddings

Recursion or embedding itself is not necessarily problematic; right-branching structures can be embedded deeply without causing parsing breakdown, as (58) illustrates. The difference between the two structures is given schematically below:

(59) *Right-branching:* $[_\alpha \ldots [_\alpha \ldots]]$

(60) *Center-embedding:* $[_\alpha \ldots [_\alpha \ldots] \ldots]$

Recursion is acceptable in (60) only to one level, and acceptable in (59) to any level. Right-branching structures are just one kind of acceptable embedding. For example, Cowper (1976) presents the following fairly complex structure involving a subject sentence with an embedded object-relative:

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

Tables 2.13–2.15 present a range of such acceptable structures. Though not all of the structures actually involve multiple embeddings, they all serve as useful constraints on theories of parsing breakdown. I will continue to refer to the class as acceptable embeddings (AE).

2.5.3 Some general parsing breakdown phenomena

In addition to listing the relevant structures, we can abstract a few important facts about parsing breakdown from the empirical studies and the collection of structures in Tables 2.11–2.15.

1. *Independence of ambiguity.* Although the definition of the parsing breakdown class given above excludes ambiguity effects, it is nevertheless an important empirical fact that local ambiguity is not necessary for parsing breakdown to occur. In particular, this cannot be the explanation for difficulty on the center-embedded constructions. It is possible to make these constructions locally ambiguous by dropping the complementizers:

   (62; PB2) The cat the bird the mouse chased scared ran away.

   (63) The cat the bird the mouse and the dog ran away.

   (64; PB1) The cat that the bird that the mouse chased scared ran away.

   Sentence (62) is locally ambiguous between a string of reduced relatives and a conjoined noun phrase (63). But the unacceptability of the construction persists even with the presence of the overt complementizers (64) (Blumenthal, 1966; Blauberg & Braine, 1974; Foss & Cairns, 1970; Larkin & Burns, 1977; Marks, 1968; Miller & Isard, 1964). Some studies have shown that structures such as (62) are *more* difficult than (64) under certain measures (Fodor & Garrett, 1967; Hakes & Foss, 1970), but the unacceptability of (64) remains firmly established.

TABLE 2.13: A collection of acceptable embedded structures (part 1 of 3).

	TYPE	EXAMPLE
AE1	Right branching (Miller & Chomsky, 1963; Kimball, 1973)	The dog saw the cat which chased the mouse into the house that Jack built.
AE2	Left branching (Kimball, 1973)	My cousin's aunt's dog's tail fell off.
AE3	Single relative clause	The man that Mary likes eats fish.
AE4	Wh-question with relative clause subject (Gibson, 1991)	What did the man that Mary likes eat?
AE5	Post-verbal center-embedded subject-relative (Eady & Fodor, 1981)	I saw the man that the woman that won the race likes.
AE6	Post-verbal center-embedded object-relative (Eady & Fodor, 1981)	I saw the man that the woman that the dog bit likes.
AE7	Post-dative-verbal center-embedded subject-relative (Eady & Fodor, 1981)	John donated the furniture that the repairman that the dog bit found in the basement to charity.
AE8	Subject sentence (Kimball, 1973)	That Joe left bothered Susan.
AE9	Subject sentence with embedded relative clause (Cowper, 1976)	That the food that John ordered tasted good pleased him.

TABLE 2.14: A collection of acceptable embedded structures (part 2 of 3).

	TYPE	EXAMPLE
AE10	Topicalization followed by subject-relative	John, the boy that the dog bit likes.
AE11	Fronted clause followed by subject-relative	While Mary slept, a sock that the dog chewed fell on the floor.
AE12	Nominalized embedded subject sentence (Kimball, 1973)	Joe's leaving bothering Susan surprised Max.
AE13	Post-verbal untensed subject sentence (Gibson, 1991)	I believe that for John to smoke would annoy me.
AE14	Post-verbal untensed subject sentence embedded in sentential complement (Gibson, 1991)	Mary held the belief that for John to smoke would be annoying.
AE15	Sentential complement with embedded subject-relative (Cowper, 1976)	The report that the armed forces that arrived first would have to stay for another year surprised me.
AE16	Sentential complement with embedded object-relative (Cowper, 1976)	The thought that the man that John liked saw the dog scared me.
AE17	Wh-question with sentential complement (Gibson, 1991)	Who did the information that Iraq invaded Kuwait affect most?

TABLE 2.15: A collection of acceptable embedded structures (part 3 of 3).

	TYPE	EXAMPLE
AE18	Post-verbal relative clause embedded in sentential complement (Gibson, 1991)	The pentagon employs many bureaucrats who the information that Iraq invaded Kuwait affected.
AE19	Post-verbal doubly-embedded sentential complement	The professor did not believe my claim that the report that the school was corrupt was biased.
AE20	Cleft with embedded relative clause (Gibson, 1991)	It was a fish that the man that Ellen married saw on the highway.
AE21	Cleft with sentential complement (Gibson, 1991)	It was the Americans that the information that Iraq invaded Kuwait affected most.
AE22	Pseudo-cleft with embedded relative (Gibson, 1991)	What the woman that John married likes is smoked salmon.
AE23	Pseudo-cleft with sentential complement (Gibson, 1991)	What the rumor that the accused man had robbed a bank influenced was the judge's decision.
AE24	Though-preposing with embedded relative (Gibson, 1991)	Intelligent though the man that Ellen married is, he has no sense of humor.
AE25	Though-preposing with sentential complement (Gibson, 1991)	Shocking though the news that Iraq invaded Kuwait was, even worse news was yet to come.
AE26	Pied-piping (Pickering & Barry, 1991)	John found the saucer on which Mary put the cup into which I poured the tea.

2. *Insufficiency of embedding depth.* Deep embeddings alone do not necessarily cause parsing breakdown. The range of constructions in Tables 2.13–2.15 make this clear[15].

3. *Fairly sharp drop in acceptability.* There is a rather sharp drop in acceptability of center-embedded structures from one level of embedding to two. Subjects almost universally judge one-level embeddings to be grammatical and two-level embeddings to be ungrammatical (Blumenthal, 1966; Marks, 1968), and performance on paraphrase tasks drops to chance levels at two levels of embedding (Larkin & Burns, 1977).

4. *Little effect of explicit instruction and training.* Subjects continue to find center-embeddings difficult after explicit instruction and training on the structures (Blauberg & Braine, 1974), in contrast to most garden path sentences. Some subjects even continue to deny that the structures are grammatical (Marks, 1968). One interesting result from the Blauberg & Braine (1974) study is that subjects were able to increase their ability to comprehend center-embeddings from one to two levels of embedding; performance on comprehension tests still dropped to chance at three embeddings.

5. *Independence of length.* Long sentences do not necessarily lead to breakdown (Schlesinger, 1968), nor do short sentences guarantee comprehensibility (PB2).

6. *Effect of semantic content.* In untimed paraphrase tasks, performance on semantically supported (SS) center-embedded sentences is better than performance on semantically neutral (SN) center-embeddings (Stolz, 1967). Examples of each type are given below:

   (65; PB1) (a) The bees that the hives that the farmer built housed stung the children (SS).
   (b) The chef that the waiter that the busboy appreciated teased admired good musicians (SN).

7. *Independence of short-term item memory.* The Larkin & Burns (1977) study demonstrated that subjects may be able to recall the words in center-embedded structures without being able to correctly pair the words (i.e, correctly parse the structure). This shows that the ability to comprehend the structure is at least partially independent of short-term memory for the words in the sentence.

2.5.4 Theories of parsing breakdown

Nearly all theories of parsing breakdown assume that structures like center-embedded relatives are difficult to comprehend because of some limit on computational resources.

[15] Surprisingly, multiply self-embedded linguistic structures may sometimes be acceptable. Schlesinger (1968) presents evidence using Hebrew texts suggesting that self-embedding of parenthetical remarks (offset by commas, not parentheses) does not necessarily lead to perceptions of ungrammaticality. However, he found that in some cases subjects claimed that unbalanced (and therefore structurally ill-formed) texts *were* grammatical, so it is a little unclear how to interpret these results with respect to parsing.

TABLE 2.16: Structural metrics for parsing breakdown.

STRUCTURAL METRIC	ACCEPTABILITY LIMIT
Ratio of nodes to terminals Miller & Chomsky (1963)	unspecified
Degree of self-embedding Miller & Chomsky (1963)	unspecified
Open sentence nodes (top-down) (Kimball, 1973)	2
Local nonterminal count (top-down) (Frazier, 1985)	8 (inferred)
Unsatisfied syntactic requirements (Gibson, 1991)	4

(Marks (1968) suggests that perhaps the structures are actually ungrammatical.) The remainder of this section explores two classes of parsing breakdown theories: *structural metrics* and *architectural theories.*

Structural metrics

Structural metrics are theories that define some metric over syntactic structures which predicts the perceived complexity of parsing the structure. Many of the theories also specify a limit beyond which structures should become unacceptable. The theories differ in the degree to which the metrics are motivated by some underlying computational architecture, but all assume, at least implicitly, that such grounding in an architecture could eventually be discovered. To the extent that the metrics are successful in predicting parsing breakdown, they can potentially help to guide the search for the architectural mechanisms. Detailed empirical critiques of each of the structural metrics may be found in (Gibson, 1991); therefore the discussion here will be kept brief.

Table 2.16 summarizes the structural metrics. For each theory, the proposed linguistic measuring unit is identified, along with the limit for acceptable structures (if specified). Some of the metrics operate on intermediate parse trees and are therefore relative to a particular parsing algorithm (strategy for enumerating nonterminals); these are noted where relevant.

Miller and Chomsky's metrics

Miller & Chomsky (1963) proposed a number of structural complexity measures, including *degree of self-embedding*, and *node-to-terminal ratio*. The self-embedding metric simply states that more deeply self-embedded structures will be more difficult to comprehend. The theory is not merely descriptive because the metric derives from a formal analysis that shows that self-embedded (as opposed to right- or left-branching) structures are precisely those structures that eventually cause trouble for any finite model of language performance. Chomsky and Miller stopped short of specifying any concrete bounds on self-embedding.

The node-to-terminal metric is the ratio of non-terminal nodes (or total number of nodes) to terminal nodes in the parse tree. This ratio provides an estimate of the amount of computation required per terminal node. As a predictor of processing breakdown, the node-to-terminal ratio fails to draw the correct contrasts between difficult center-embeddings and other acceptable embeddings (Gibson, 1991). This is not too surprising because Chomsky and Miller clearly did not intend for the metric to account for difficulty on center-embeddings, since they had proposed the independent self-embedding metric. Nevertheless, nearly all subsequent theories of processing breakdown have adopted some form of either the finite state explanation or the non-terminal/terminal ratio.

Principle of Two Sentences

Kimball (1973) proposed that human parsing proceeds top down, and no more than two sentence nodes can be parsed at the same time. This rules out doubly-embedded relatives like (57) since three S nodes must be open (the main clause and the two subordinate clauses). This principle accounts for the unacceptability of a number of other constructions as well, such as embedded subject sentences (PB5). Surprisingly, however, the principle overpredicts parsing breakdown, as (61) above demonstrates (AE9). Upon encountering the relative clause *that John ordered*, a top-down parser has three open S nodes: the main clause, the subject sentence (*that the food tasted good*), and the embedded relative. Nevertheless, the sentence does not produce the breakdown associated with doubly-embedded object-relatives.

Maximal local nonterminal count

Frazier (1985) modified Chomsky and Miller's original node-to-terminal metric so that it is a local rather than global measure, in an attempt to better capture moment-by-moment processing difficulty. A local nonterminal count is the count of nonterminal nodes introduced while parsing a short segment of the input stream. Frazier defined a short segment as three adjacent terminals. The maximal local nonterminal count is the largest local nonterminal count that occurs during a sentence. Frazier assumed that S nodes counted as 1.5 while all other nonterminals counted 1. The prediction is that sentences with high local nonterminal counts will be more difficult to process than sentences with low counts. Figure 2.4 shows two examples of how the metric is computed.

Frazier examines a number of subject sentence and center-embedded constructions to support the metric. This is the first metric which correctly predicts the contrast between center-embeddedings in object position vs. subject position (PB1 vs. AE6, (Eady & Fodor, 1981).) However, as Gibson (1991) points out, the metric fails to account for the basic finding that doubly-embedded relatives are more difficult that singly-embedded relatives:

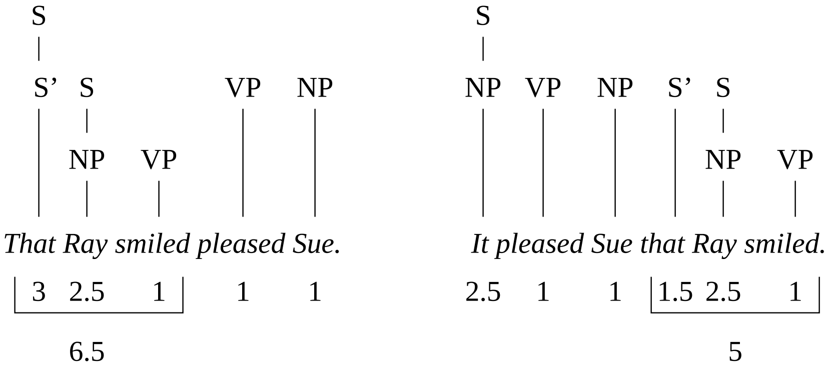


FIGURE 2.4: Computing maximal local nonterminal counts. The locality of the metric is three adjacent terminal nodes. S nodes count as 1.5, all other non-terminal nodes count as 1.

(66) (a) The man that the dog bit ate the cake that the woman saw.
(b) The man that the woman that the dog bit saw ate the cake.

Frazier's metric incorrectly assigns the same value to (66a) and (66b). The fundamental problem is that the metric predicts difficulty only when there is a high density of nonterminals introduced over a short span; specifically, when three high complexity words (in the sense that they produce a high nonterminal count) are immediately adjacent. This may be the case for center-embedded structures with dropped complementizers (PB2), of the kind Frazier examined, but parsing breakdown may arise even when the nonterminals are somewhat more evenly distributed across the sentence, as in (66a).

Gibson's overload metric

Gibson (1991) developed a detailed metric within the GB framework that attributes a cumulative cost to maintaining structure with locally unsatisfied syntactic requirements. In particular, the theory assigns a cost to maintaining thematic-role bearing elements which have not yet received their thematic roles, and to lexical projections which do not yet have their lexical requirements satisfied. (67) shows the structure produced by Gibson's parser upon processing the third NP in a center-embedded construction:

(67; PB2) (a) The man the woman the dog bit likes eats fish.
(b) [$_{IP}$ [$_{NP}$ the man$_i$ [$_{CP}$ [$_{NP}$ O_i] [$_{NP}$ the woman$_j$ [$_{CP}$ [$_{NP}$ O_j] [$_{IP}$ [$_{NP}$ *the dog*]]]]]]

There are five NPs (three lexical, two nonlexical) that require thematic roles but lack them. The theory states that five such local violations (which may involve syntactic requirements other than thematic role assignment) is enough to cause processing breakdown, while four is acceptable. Gibson demonstrates that the metric accounts for nearly all of the parsing breakdown effects in Tables 2.11–2.12. This was a major empirical advance over previous theories, which were primarily concerned only with center-embeddings.

TABLE 2.17: Architectural theories of parsing breakdown.

ARCHITECTURE	LIMITED RESOURCE	PROPOSED LIMIT
Push-down automaton (Yngve, 1960)	Stack cells	7
Subroutine architecture (Miller & Isard, 1964)	Return address memory	1
ACT (Anderson et al., 1977)	Control variables	1
Sausage Machine (Frazier & Fodor, 1978)	Lookahead window	6 words
YAP finite state machine (Church, 1980)	Stack cells	unspecified
Register-vector FSM (Blank, 1989)	State for tracking clause level	3
Unification Space architecture (Kempen & Vosse, 1989)	unspecified	unspecified
PDP network (Weckerly & Elman, 1992)	Hidden units	unspecified

Comprehension/parsing architectures

Architectural theories define the conditions for parsing breakdown in terms of some specific computational architecture. Unlike the structural metrics, the relationship to an architecture is inherently specified as part of the theory. While this is a clear theoretical advantage over the structural metrics, no architectural theory yet proposed comes close to the coverage of Gibson's metric.

Table 2.17 summarizes the architectural theories. For each theory, the relevant computational resource or mechanism is identified, along with the proposed limit on that resource (if specified).

Yngve's depth metric

Yngve's (1960) model was the first attempt to develop a well-specified computational account of structural complexity. The model is essentially a push-down automaton (PDA) that generates phrase structure trees in a top-down, left-to-right manner. Although it was originally intended as a model of sentence production, the automaton can be used for parsing as well. The restriction Yngve imposed on the model was to limit the stack depth to seven, a value motivated by Miller's famous theory of short term memory (Miller, 1956).

There are a number of empirical problems with the model, the most serious being that it predicts unbounded left-branching (AE2) to be as unacceptable as center-embedding—a consequence of the purely top-down algorithm (Miller & Chomsky, 1963). This is a serious problem because, although English is predominantly right-branching, there exist languages (e.g., Japanese) that are predominantly left-branching.

Despite the empirical problems, Yngve's model has a combination of theoretically desirable features that many later models do not share: it is instantiated in a well-defined computational architecture, it makes clear and precise predictions on any given structure, and the architectural limitations have some possible grounding in independently developed psychological theory (namely, the Miller 7+/-2 theory).

Subroutine architecture

Miller & Isard (1964) suggested that embedded clauses may be processed by calling a subroutine for parsing clauses. If there is a severe limit on the number of return addresses that may be stored, then the processor will encounter difficulty with self-embeddings. If only one return address may be stored, then this would account for the difficulty on doubly-embedded relatives. A form of this hypothesis showed up later in (Anderson et al., 1977) and (Blank, 1989), discussed below.

ACT

Anderson, Kline and Lewis (1977) developed a model of language comprehension within the ACT theory of cognition. Since ACT's procedural component is a production system, this required specifying the set of productions used in parsing and interpretation. The control of the parsing productions is accomplished via a set of control variables that maintain the state necessary to handle multiple embeddings. These variables permit control to be returned to productions parsing higher level clauses after embedded clauses are complete. The model has enough control variables to handle only one level of embedding, so breakdown occurs with double embeddings. The model is essentially an instantiation of the (Miller & Isard, 1964) theory of subroutine interruption. Unbounded right branching as in (58) is not problematic since control need never return to the main clause productions. Anderson et al. acknowledged that the limitation is essentially arbitrary, but pointed out that any *unbounded* memory of state would have to be maintained in ACT's semantic network. Since this network is not completely reliable, breakdown would eventually occur.

The Sausage Machine

Frazier & Fodor (1978) attribute the difficulty with center-embeddings to problems that the PPP has in performing the initial phrasal segmenting. They assume that the initial adjacent noun phrases or the terminal verb phrases will be incorrectly interpreted as conjoined phrases by the PPP, which must make its structuring decisions without access to the parts of the sentence that fall outside the six-word window. However, there are difficult center-embeddings which fall completely within the window:

(68; PB2) Women men girls love meet die.

Since the PPP is able to generate S nodes on its own, (68) should not cause it difficulty. Actually, the theory Frazier and Fodor propose is more complex than this, since they suggest that the PPP may be garden-pathed even though all the relevant information is available to it in the window. This part of the theory was not worked out in much detail and begins to undermine the whole approach of having a restricted window.

YAP finite state machine

Church's (1980) YAP is an implemented system that combines the determinism hypothesis of Marcus (1980) with the idea that the human parser must be a finite state machine with a rather limited amount of state. The architecture is essentially that of PARSIFAL, with the exception that the stack is bounded. Unlike Yngve (1960), however, Church did not venture to propose what the limit on the stack might be, except to note that it must be fairly shallow given the difficulty on center-embeddings.

Register vector machine

Blank's (1989) parser (introduced earlier) maintained state in a fixed-length vector of three-valued syntactic features and control variables. Part of the vector is devoted to keeping track of what clausal level is currently being parsed. The vector only has enough state to track three levels (main clause, embedded clause1, and embedded clause2), so it is unable to parse triply center-embedded relatives. Indefinite right- or left-embedding does not invoke clause shifting and the parser handles these easily. As specified, there seems to be enough state to parse even difficult doubly-embedded relatives. If the state vector was restricted to better match human performance, the architecture would make the same predictions (both correct and incorrect) as Kimball's (1973) Principle of Two Sentences.

Unification Space architecture

Kempen & Vosse (1989) developed and implemented a novel computational architecture based on activation decay and simulated annealing. The system works by retrieving syntactic segments from a lexicon, and then attaching the segments in a stochastic process that favors more highly activated nodes. The segments consist of nodes linked by functional syntactic relations; attachment occurs by unifying individual nodes. Since activation decays, more recent nodes are more active. The temperature of the annealing process is a function of the total activation in the system, so the process gradually "cools". Kempen and Vosse present results showing that the system parses singly-embedded relative clauses correctly about 98% of the time, but parses doubly-embedded relatives only about 50%

of the time. The corresponding results for singly and doubly-embedded right-branching structures are 97% and 82%, respectively.

Although this is an extremely interesting result, Kempen and Vosse offer no immediate explanation for the performance difference between the two structures; it is unclear whether this is truly the first nonparametric theory of center-embeddings, or whether there is some variability in the basic architecture which could lead to different results.

Weckerly and Elman's PDP model

Weckerly & Elman (1992) constructed a connectionist system that models some aspects of human performance on center-embeddedings. As the network is given words one at a time, it encodes the content of the words and the surface order, in a bottom-up parse. Since there is a fixed amount of structure devoted to encoding this state, the information in the network eventually degrades as the state gets large. After processing the three initial noun phrases in a doubly center-embedded sentence, the ordering information is lost and the parse fails. However, the lack of order information can be compensated in the network by semantic constraints, so that in semantically supported sentences (see §2.5.3), the network can still manage to give the correct output. This is the first model of processing center-embedded structures that begins to account for semantic effects. However, it remains to be seen how the model will scale to handle the range of effects listed in Tables 2.11–2.15.

Finite state and self-embedding

The key feature that most of these models have in common (particularly the models of Anderson, Blank, Chomsky and Miller, Kimball, Wanner, Weckerly and Elman, and Yngve) is a commitment to finite state. There are two good reasons for this:

1. Any finite machine will eventually fail to recognize self-embedded structures. In other words, self-embedding grammars fall outside the computational scope of finite automata; equivalently, no regular grammar is self-embedding. (The proof involves the pumping lemma; see, for example (Lewis & Papadimitriou, 1981)).

2. Chomksy (1964) proved that it is *only* self-embedding that causes languages to be context-free. More precisely, if L is a context-free language, then it is not a finite-state language if and only if all of its grammars are self-embedding.

Thus, simply adopting a finite state model makes precisely the right cut with respect to center-embeddings: finite state machines will always eventually fail to recognize some center-embedded structures, while at the same time they are in principle capable of recognizing *all other* kinds of structures produced by context-free grammars (e.g., indefinite right-branching or left-branching structures).

2.6 Summary of the phenomena

This section briefly summarizes the major findings concerning the sentence processing phenomena discussed in this chapter, and closes with a discussion of how the phenomena together provide great mutual constraint for any comprehension theory.

The products of comprehension. Comprehension produces a syntactic, semantic (intensional), and referential (extensional) representation. There are functional reasons for all three representations. The functional analysis is further supported by actual practice in building working systems. The existence of a referential representation is empirically supported by experiments that demonstrate the confusability of referentially and inferentially equivalent expressions. These same experiments show that in some cases the final memory for text is primarily referential. On the other hand, the independence of a semantic representation is supported by experiments demonstrating that in some cases memory for text may be primarily semantic. The nature of the memory is influenced by factors such as the difficulty in constructing a mental model. A mental model is a referential representation that represents one particular situation at a time, and maintains a direct correspondence between the elements in the representation and the elements in the domain. Evidence for this form of representation comes from experiments contrasting models with more powerful alternatives (logic) on a variety of reasoning tasks.

Immediacy of interpretation and the time course of comprehension. The referential, semantic, and syntactic representations are computed immediately and incrementally on a word-by-word basis. In reading, the time course of this processing ranges from 50-1000+ ms per word. The evidence for immediacy comes from a wealth of experiments using speech shadowing, eye movement, and cross-modal priming techniques. There have been suggestions that some kinds of syntactic information is systematically delayed (such as verb subcategory), but thus far the evidence weighs in favor of universal syntactic immediacy. There are limits on the immediacy of reference resolution and mental model construction. Although the processes are immediately initiated, completion may be delayed due to the structure of the text itself, or computational limitations of comprehension. A number of experiments have provided evidence for the distinction between automatic, or on-line model construction, and cognitive, or deliberate model construction. The depth and nature of processing depends on a number of factors such as time available and the goals in comprehension.

Structural ambiguity resolution. Both on-line ambiguity resolution and final preferred interpretations may be influenced by structural, lexical, semantic, and contextual factors. No single principle or preferences, or class of preferences, has been found to universally characterize ambiguity resolution. A large number of empirical studies show that both modular and interactive effects may arise across a wide variety of contexts, structural ambiguity types, and experimental paradigms. Of the major parsing preferences proposed, some combination of Right Association and lexical preferences has been found to be the most robust in studies of natural corpora. Although there is some evidence for a limited amount of structural parallelism in the parsing process, the paradoxical results of some of the experiments makes interpreting the evidence difficult at this time.

Garden path effects and unproblematic ambiguities. GP effects sometimes arise when a reader or listener misinterprets a local ambiguity and cannot easily recover the correct interpretation. The result is an impression of ungrammaticality. There are a wide variety of structures associated with GP effects. The evidence comes from a range of experiments and informal surveys using grammaticality judgment tasks. Complementing the GP structures are an equally rich set of unproblematic ambiguities, which do not cause difficulty no matter which interpretation of the local ambiguity proves correct. GP effects are recoverable in that a GP sentence may be perceived as grammatical once the correct interpretation is discovered. GP effects are generally independent of length (though some distance-to-disambiguation effects have been detected), lexical ambiguity, semantic content, and the assumed preferred interpretation of a given ambiguity (i.e., GP effects may be bidirectional).

Parsing breakdown and acceptable embeddings. Parsing breakdown occurs when a listener or reader is unable to comprehend and perceive a sentence as grammatical without great difficulty. Parsing breakdown technically includes GP effects, but parsing breakdown may occur independently of ambiguity. Breakdown on unambiguous center-embedded structures has been demonstrated using a range of measures, including grammaticality judgments, recall tasks, paraphrase tasks, and question answering. There is a fairly sharp drop in acceptability from one center-embedded relative clause to two. A variety of structures causing parsing breakdown have been discovered (though none have the thorough empirical backing that center-embeddings do). Complementing the PB structures are a variety of acceptable embeddings such as right-branching, which may be iterated indefinitely. In contrast to GP effects, instruction and practice have only marginal impact on the acceptability of difficult structures. Semantically constrained material does boost performance on untimed paraphrase tasks. PB effects are independent of length, and also independent of short term memory of the words in the sentence.

Figure 2.5 gives a directed graph summarizing the constraining properties of the phenomena. The graph should be interpreted as follows: $X \rightarrow Y$ means that phenomena X constrains the theoretical explanation of phenomena Y. Each arc is explained below (the arcs are labeled with lower case letters).

(a) The mechanisms explaining garden path effects must not be too weak that they fail to account for unproblematic ambiguities. Likewise, the mechanisms explaining unproblematic ambiguities must not be so powerful that they fail to account for garden path effects.

(b) Immediacy of interpretation constrains the explanation of unproblematic ambiguities by ruling out certain kinds of lookahead or delayed commitment comprehenders.

(c,d) The phenomena surrounding ambiguity resolution constrain GP/UPA theories which consist in part of principles for guiding the initial interpretation.

(e) Models of ambiguity resolution must be consistent with immediacy of interpretation.

(f) Evidence for structural, lexical, semantic, and contextual effects all constrain theories of ambiguity resolution.

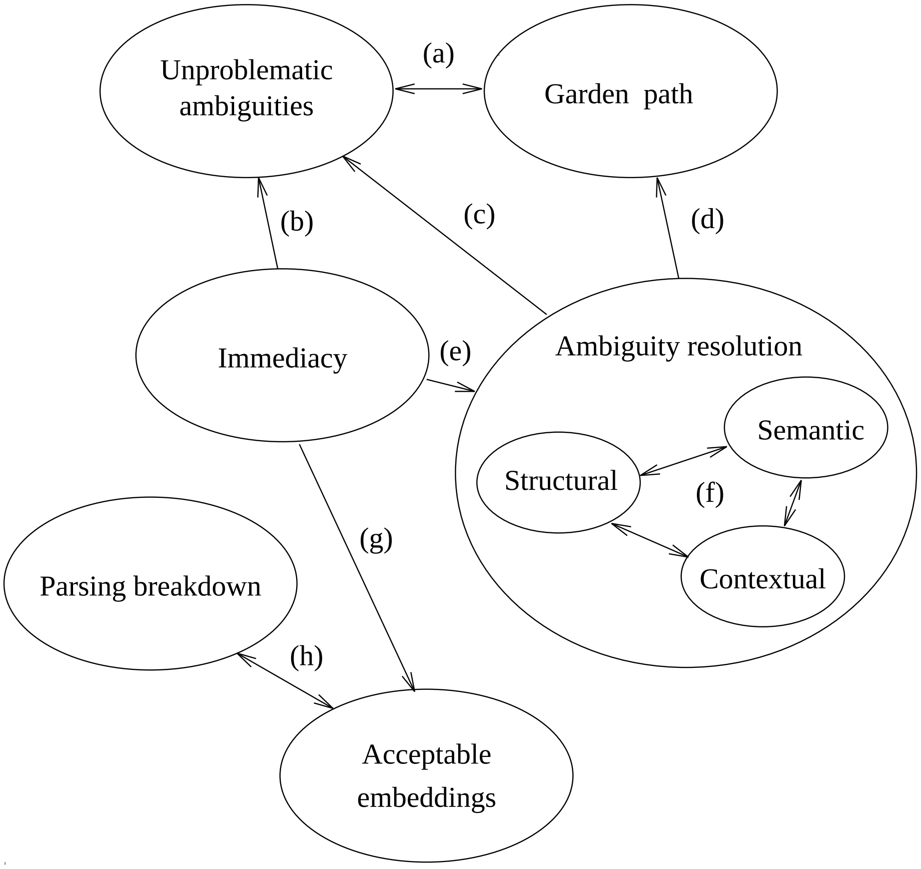


FIGURE 2.5: How the phenomena mutually constrain the theory. $X \rightarrow Y$ means that phenomena X constrains the theoretical explanation of phenomena Y. Each arc is explained in the text.

(g) The mechanisms for handling acceptable embeddings must be consistent with immediacy of syntactic parsing.

(h) The mechanisms explaining parsing breakdown must not be so weak that they fail to account for the acceptable embeddings. Likewise, the mechanisms that handle the acceptable embeddings must not be so powerful that they fail to predict parsing breakdown.

Chapter 3

The NL-Soar Theory

There seems no way to enter the fray with a little theory and/or a highly approximate one. To do so, is to invite the wrath of the linguistic gods. Full blown from the brow of Zeus or nothing!
— Allen Newell

To criticize the pages on language that follow would be like shooting fish in a barrel.
— Angry linguistic god[1]

THIS CHAPTER DESCRIBES THE NL-SOAR COMPREHENSION MODEL built within the Soar architecture. The first section lays the necessary foundation by examining the nature of cognitive architectures generally, reviewing the Soar architecture, and establishing the NL-Soar approach of studying language and architecture. The core of the chapter describes the model itself, along with examples illustrating its operation. We then step back and explore the space of potential Soar comprehension models, to motivate some of the major design choices in the current NL-Soar. The chapter concludes with a summary of the theory.

3.1 Preliminaries: architectures and Soar[2]

Because this thesis purports to present an *architecturally-based* theory of comprehension, it is important to explain exactly what that means and why it is a desirable aspect of the theory. The explanation that follows can be taken as part of the answer to the question: Isn't Soar just a programming language used to *implement* NL-Soar? We shall see that it is far more than that, and we will take up the issue again in Chapter 9, considering there the broader issue of Soar's role in the theory and its empirical coverage.

[1] Derek Bickerton, book review of *Unified Theories of Cognition* (Bickerton, 1992).

[2] Parts of this section are based on an unfinished manuscript that I was working on with Allen Newell in the Spring of 1992. Any misconceptions or errors that remain are of course entirely my responsibility.

3.1.1 What is a cognitive architecture?

A cognitive architecture is the (relatively) fixed computational structure that supports cognition (Newell, 1990). An architecture specifies the available *processing primitives* (operations on data), *memory structures* (support for storing and retrieving encoded knowledge), and *control structure* (specification of how processing unfolds). Architectures are universal—they are just those computational structures that admit programs. For behavior to emerge, both the architecture and the content (the program) must be specified.

The central role of architecture in cognitive science

The central tenet of cognitive science is that cognition is a kind of computation. If cognition is computation, there must be an architecture to support it. In this view, discovering the nature of mental architecture is the most fundamental question in cognitive science, for a theory of mind must *be* an architectural theory (Anderson, 1983; Newell, 1990; Pylyshyn, 1984).

This has significant impact on how we construct and evaluate cognitive models. If a cognitive model is to make the strong claim that the processing steps in the model correspond to the processing steps in the human, then the model must incorporate some assumptions about the underlying architecture, because it is the architecture that defines the nature of the available processing primitives (Pylyshyn, 1984). Pylyshyn calls this form of correspondence *strong equivalence*.

Making explicit architectural assumptions also helps to clarify what carries theoretical content in implemented cognitive models. For example, if a Turing Machine was seriously proposed as a theory of mental architecture, then a cognitive model could be constructed for some particular task by developing a computer system that forms a virtual Turing Machine architecture and then programming that architecture with the program and data relevant to the task. The particular implementation of the Turing Machine—whether it is coded in Lisp or C, on a parallel machine or serial processor—is irrelevant. What is theoretically significant is the architecture itself and the content posited to produce the task behavior.

3.1.2 The Soar architecture

This section provides a brief overview of the essentials of the Soar architecture. Although Soar was first described as an artificial intelligence system (Laird, Newell & Rosenbloom, 1987) , it emerged as a theory of the human cognitive architecture in Newell's 1987 William James lectures (Newell, 1990). Soar has since been applied to a wide range of cognitive tasks (Lewis, et al., 1990; Rosenbloom, Lehman & Laird, 1993). For more complete overviews of Soar and Soar research, see Chapter 4 of (Newell, 1990), the recent Soar6 manual (Laird, Congdon, Altmann & Doorenbos, 1993), and the recent edited collection of papers (Rosenbloom, Laird, & Newell, 1993a).

Fundamental components

The basic components of Soar are shown in Figure 3.1[3]. All behavior, from routine to difficult, occurs in *problem spaces*, shown as triangles in the figure. A problem space is a formulation of a task as an *initial state*, a *goal state*, and a set of *operators* that apply to states and produce new states (Newell & Simon, 1972; Newell, 1980; Newell, 1990). Any application of operators that yields the goal state is taken as a solution to the problem. A working memory holds the momentary problem solving *context*: aspects of the problem space, state, and operator are represented as declarative attribute-value structures. Knowledge about how to apply and select operators and problem spaces is held in a long-term *recognition memory*, which continually matches in parallel against the declared context. The recognition memory consists of a large[4] set of condition-action associations (productions). The conditions specify patterns that occur in working memory, and the actions retrieve knowledge in the form of *preferences* to change aspects of the problem space context. All long-term knowledge, whether declarative or procedural, is held in the uniform recognition memory.

A step in the problem space (e.g., an operator application, or an initial state selection) is taken each *decision cycle* (bottom of figure). The decision cycle consists of two phases. During the *elaboration* phase, recognition memory matches against working memory, and associations fire in parallel and in sequence until *quiescence* is reached, that is, until all the relevant associations have finished firing. At quiescence, the retrieved preferences are interpreted by the *decision procedure*, which determines the next step in the problem solving. The decision procedure simply implements the semantics of a fixed preference language, which allows a partial ordering to be defined over problem space alternatives.

If the retrieved preferences uniquely determine the next step to take, the decision procedure effects that step. In such a case Soar proceeds by recognition. But this need not be the case; knowledge may be inconsistent, inconclusive, or missing, causing an *impasse* to arise after quiescence. Soar responds to impasses by setting up a new problem space in which to deliberately acquire the necessary knowledge. Impasses may occur on any problem space function. For example, Figure 3.1 shows two impasses: the top impasse is due to lack of knowledge to select among a set of operators, and lower one is due to lack of knowledge to apply an operator. Impasses may occur indefinitely, leading to a *subgoal* hierarchy in working memory. Any impasse (not just the last one) may be resolved at any time, resulting in an automatic collapse of the subgoals.

As knowledge from a lower problem space is accumulated to resolve a higher impasse, Soar's learning mechanism, *chunking*, builds new associations in the recognition memory that will retrieve the knowledge in relevant contexts by match. Thus, in future similar

[3] The description of Soar in this Chapter mixes problem space and symbol-level mechanisms; there is a growing view that a *problem space computational model* (PSCM) can be described independently of particular symbol-level implementations (Newell et al., 1991). The PSCM is essentially a specification that mixes abstract symbol-level and knowledge-level components. For the purposes of this thesis, the more traditional description of Soar will suffice.

[4] The largest Soar system contains over 300,000 productions (Doorenbos (1993); Bob Doorenbos, personal communication), though most Soar systems are still less than 5,000 productions.

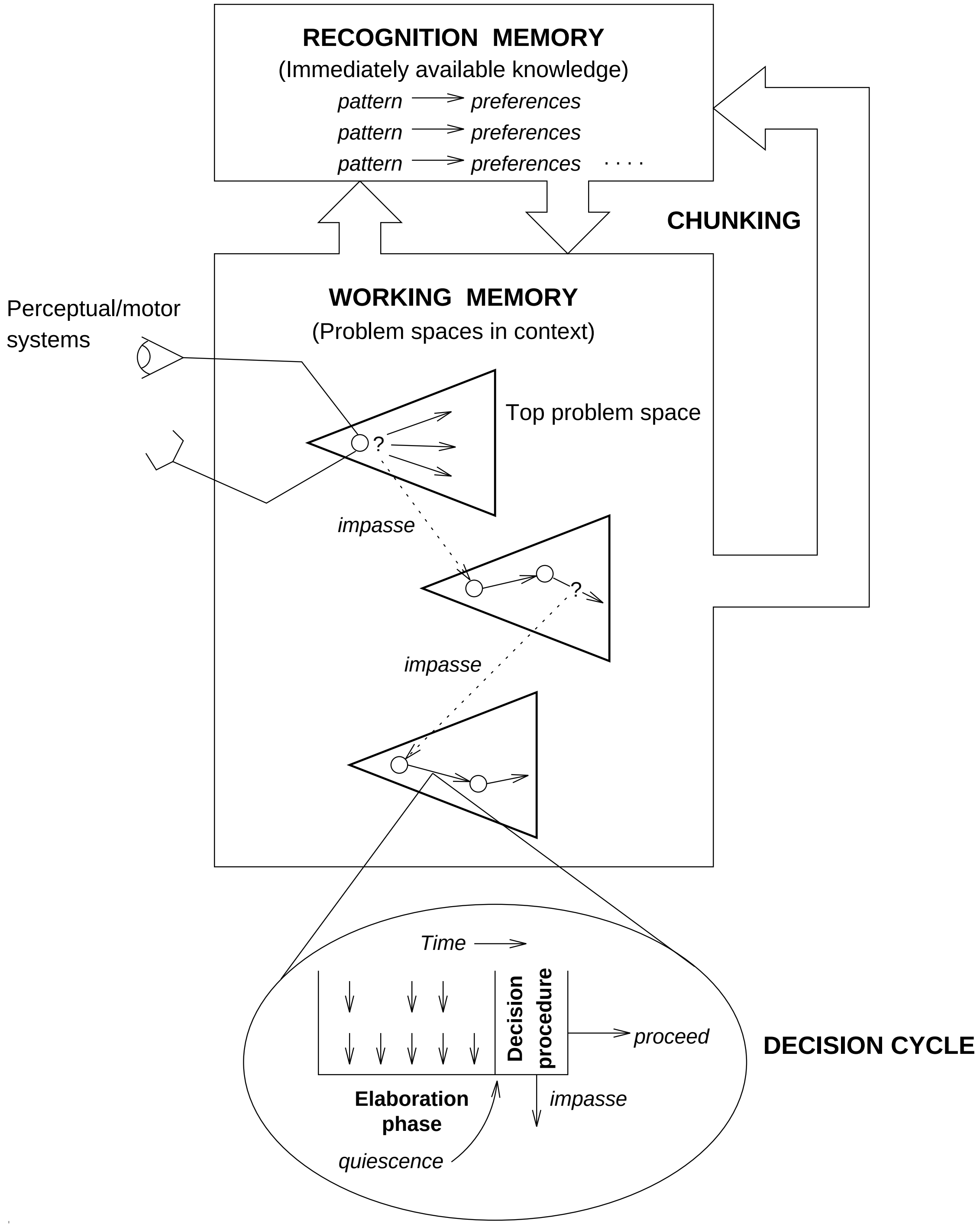


FIGURE 3.1: The Soar architecture. The three triangles represent active problem spaces in working memory. The small circles within the triangles represent states; the arrows within the triangles represent operators. The downward pointing arrows in the elaboration phase represent parallel (vertically stacked) and serial association firings.

situations, the impasse and the deliberation it leads to may be avoided. Chunking is part of the architecture: it happens continuously and automatically. Although there is only one learning mechanism in Soar, chunking works over every kind of impasse in every kind of cognitive activity, giving rise to many kinds of learning (Steier et al., 1987; Newell, 1990).

Memory, process, and control

As with any functionally complete architecture, we can identify the memory, process primitives, and control structure of the Soar architecture. *Memory* in Soar consists of the large recognition memory and the declarative working memory[5]. Both are unbounded memories, but with very different properties. The knowledge held in recognition memory is only retrieved if the appropriate cues are present in working memory. The associations cannot be modified or removed once added, nor are they examinable by other processes. The burden of providing free composability rests on the working memory, which can be rapidly deployed by the system as it sees fit.

The *processing primitives* in Soar are the basic operations available to affect memory. For the working memory, these include operations to add and delete new attribute-value structures. For the recognition memory, the operations are matching and chunking. However, unlike the working memory primitives, the operations on recognition memory are not modulated by knowledge; that is, their evocation does not depend on the contents of memory. (The *results* of the match process do, of course, depend on the contents of working memory.) Learning and match are continuous, automatic processes. No retrieve or store operations appear in the actions of associations.

Soar's *control structure* is a combination of the recognition match and the decision cycle. Each processing step depends on the preferences retrieved at that moment by the condition-action pairs in recognition memory. The fixed decision cycle processes these preferences to determine the next problem space step. The control is therefore *open*, since all the knowledge in long-term memory is brought to bear at each point, and the control memory is open to addition (by chunking). This *recognize-decide-act* structure contrasts with the *fetch-decode-execute* cycle of standard computer architectures, which restricts the active control memory to a local piece of program fixed before execution.

Perception and action

Soar interacts with the outside world through perceptual and motor modules that make contact with central cognition via working memory, specifically, through the state of the top problem space (Figure 3.1.2). Autonomous *encoding* productions parse perceptual input to prepare it for cognition; similarly *decoding* productions prepare motor commands for direct execution (not shown in the figure). Encoding and decoding productions are autonomous in

[5] There are actually other memories in Soar, including the preference memory to hold the retrieved preferences to be interpreted by the decision cycle, and a memory to hold partial match information for the production system.

TABLE 3.1: The time scale of Soar processes.

ARCHITECTURAL PROCESS	COGNITIVE FUNCTION	TIME SCALE
Search in problem space(s)	Unit task	∼∼10 s – minutes
Operator implemented in subspace	Simple composed operation	∼∼1 s
Decision cycle	Elementary deliberate act	∼∼100 ms
Recognition memory match	Distal knowledge access	∼∼10 ms

that they do not depend on the current problem space context, and they fire independently of the decision cycle.

The perceptual/motor side of Soar is presently underdeveloped, though it is an active area of research (e.g., (Wiesmeyer, 1992)). The actual implementation lags behind the model presented in (Newell, 1990); in particular, the encoding/decoding scheme has not been implemented. For the purposes of this thesis however, the details of the perceptual-motor system will not play a major role.

The temporal mapping of Soar

Newell (1990) provides an analysis that grounds Soar temporally as a model of human cognition. The results are summarized in Table 3.1. The analysis is constrained from above by the functional requirement to yield cognitive behavior in about a second, or a few hundred milliseconds in the most elementary reaction tasks. The analysis is constrained from below by basic temporal properties of neural circuitry: distal processing (beyond local circuits of about a cubic millimeter) cannot happen faster than roughly 10 ms, since the characteristic processing time of local circuits is ∼∼1 ms. The elementary function of distal access in Soar is provided by the recognition match, so the recognition match must take on the order of 10 ms. There is only room for two more system levels between distal access and cognitive function, corresponding to the decision cycle and composed operator in Soar. Fortunately, both the bottom-up and top-down analyses yield consistent results for Soar. As Newell repeatedly emphasized, these values should not be taken as precise values, but rather order-of-magnitude estimates (Newell used the notation ∼∼100 to mean 30-300). Wiesmeyer (1992) and Johnson-Laird (1988) use a constant of 50 ms per operator to make quantitative predictions across a range of immediate response tasks.

Coverage of Soar as a psychological theory

Soar has been applied to a wide range of cognitive phenomena, including immediate reaction tasks, classic problem solving puzzles (e.g., towers of Hanoi), verbal reasoning (e.g., syllogisms), and repetitive practice effects. Newell (1990) is still the most comprehensive reference; Lewis et al. (1990) and Rosenbloom et al. (1993b) provide more recent summaries.

3.1.3 Language and architecture

How does language comprehension (and production, and acquisition) fit into an overall theory of mental architecture? This is the modern computational version of discovering the relationship between language and thought.

Most psycholinguistic work does address architectural issues to some extent. For example, distinctions are drawn between between modular and interactive architectures, or automatic and cognitive processes. But there have been relatively few explicit proposals of functionally complete architectures for language processing[6]. Exceptions include Marcus's (1980) PARSIFAL, the CAPS architecture underlying CC READER (Just & Carpenter, 1992), the PDP architecture of St. John & McClelland (1990), and the annealing/activation-based architecture of Kempen & Vosse (1989). All of these models make explicit assumptions about the control structure, processing primitives, and computational resources supporting linguistic behavior. (The Sausage Machine (Frazier & Fodor, 1978) was a step in the architectural direction, but was never specified in much detail).

Fewer still are those theories that relate the architecture of language processing to the architecture of cognition generally. This is largely a result of an assumption in most theorizing that linguistic processing is modular (Fodor, 1983). Thus, when explicit architectural hypotheses are made, there is often no attempt to generalize them beyond language. A notable exception is the work of Carpenter and Just—their CAPS architecture forms the basis of a general cognitive theory that has been applied to some nonlinguistic tasks (Carpenter et al., 1990).

Modularity defines the first choice to be made in developing a Soar theory of language comprehension: should a special linguistic input system be posited outside of the existing architecture? Or should the comprehension capability be developed within the given mechanisms? If we take what appears to be the modular route, we essentially start fresh with respect to defining the control structure, processes, and memories for comprehension. The interesting issue then becomes the nature of the interface between Soar and the linguistic module.

The NL-Soar approach

The alternative approach, and the one we have adopted with NL-Soar (Lehman, Lewis & Newell, 1991a, 1991b, 1993; Lehman, Newell, Polk & Lewis, 1993; Lewis, 1993a, 1993b; Lewis, Newell, & Polk, 1989; Newell, 1987, 1990), is to embed the comprehension capability within the existing architecture. This is the path urged by Newell in the William James lectures, where he first sketched the Soar theory of comprehension. It is essentially a minimalist approach, attempting to see how far the present mechanisms can be pushed

[6] In fact, the situation seemed serious enough to Forster (1979) that he issued a general call for more complete information processing theories of comprehension—in effect, Forster was encouraging architectural theories, though he did not use those terms. Forster's concerns for psycholinguistics quite closely paralleled Newell's concerns for psychology in general, as expressed six years earlier in the famous 20 questions paper (Newell, 1973b).

before positing new ones. Newell clearly viewed the success or failure of the venture as an open question, with no a priori resolution. It could in fact turn out that Soar is not up to the phenomena of real-time language comprehension.

Such an approach may seem to be completely at odds with the modularity hypothesis, and, more to the point, with the evidence accumulated in favor of it. Real-time processing, automaticity, etc., may seem to be thrown out in favor of a view of language as general problem solving. Prejudging the approach in this way is misguided, however. Even in a modular architecture, there must exist some relationship between linguistic and nonlinguistic processing, and there may even be architectural principles in common across the modules. The apparently non-modular research path we have taken is one way of discovering these commonalities. More importantly, the remainder of this thesis should make clear that the basic phenomena of real-time comprehension are dealt with in considerable detail, and the approach has led to an even richer understanding of modularity than might otherwise have been possible (Chapters 4 and 9).

3.2 The basic structure of NL-Soar

Building a comprehension model in Soar requires specifying the problem spaces and operators that achieve the functions of comprehension. This section lays out the basic structure of NL-Soar in these terms. The first order of business, therefore, is not describing NL-Soar along traditional dimensions of parsing, such as top-down or bottom-up or left-corner, but rather specifying how comprehension is realized in Soar's architectural mechanisms. Of course, the traditional characterizations are both possible and useful—but they are only part of the story.

3.2.1 Comprehension operators and the real time constraint

Soar comprehends language by applying *comprehension operators* to the incoming linguistic input. These operators produce syntactic, semantic, and referential representations in working memory (§2.1). Functionally, comprehension operators accomplish the mapping

$$L \times U \times S \times R \times P \rightarrow U \times S \times R$$

where L corresponds to possible incoming linguistic input; U, S, and R correspond to possible syntactic, semantic, and referential representations, respectively; and P corresponds to the current problem solving context. (U refers to the $\underline{\text{u}}$tterance representation, to be explained in a moment).

Immediacy requires that the comprehension operators effect this mapping incrementally over a small grain size of linguistic input (at least at the word level). Given the Soar temporal mapping (§3.1.2), this incremental mapping must occur with just a few operators per word. Let us use the 50 ms constant that serves as the basis for Soar's chronometric predictions in immediate reaction tasks. A comprehension rate of 250 words per minute, or 240 ms per word (§2.2.3) means that, on average, comprehension must complete its work in about

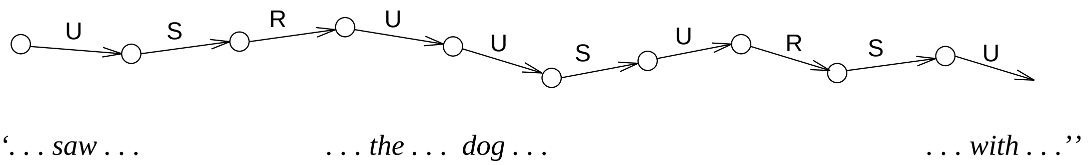


FIGURE 3.2: Recognitional comprehension. A stream of comprehension operators applies to the incoming linguistic input. There are three types: syntactic operators (U), semantic interpretation operators (S), and reference resolution operators (R).

four or five operators. This is the first and most serious constraint on the NL-Soar model. It means that comprehension must proceed mostly *recognitionally*, in other words, without incurring impasses and engaging in deliberate problem solving. The knowledge to propose, select, and implement the comprehension operators must be immediately available via the recognition memory match.

3.2.2 The structure of comprehension operators

Even with the tight constraint provided by real-time immediacy, there are a number of alternatives for structuring comprehension operators, corresponding to different ways of distributing knowledge across the various problem space functions (operator proposal, operator selection, and operator application). We will consider this space later in §3.7, but for now simply posit the following three kinds of comprehension operators:

- *U-constructors* build the *utterance model*, which represents the syntactic structure of the utterance.
- *S-constructors* build the *situation model* which represents the meaning of the utterance.
- *Resolve operators* perform reference resolution by recognizing parts of the situation model as descriptions of previously mentioned or known entities, and elaborating the situation model with that information.

Given these types, Figure 3.2 shows an example of what recognitional comprehension looks like as a stream of comprehension operators. Notice that every word need not evoke all three operator types, that more than one operator of a given type may apply per word, and that there is no fixed ordering of application. All that this scheme assumes is that the set of operators is sufficient for incrementally constructing the comprehension data structures, with some division of labor among the different types. Furthermore, it must be the case that these operators must rapidly bring to bear multiple knowledge sources, if they are to accomplish the required mapping in a just a few operators.

3.2.3 From deliberation to recognition: comprehension as a skill

To reiterate, achieving the purely recognitional comprehension illustrated in Figure 3.2 requires that the proposal, selection, and application of the comprehension operators be accomplished directly by associations in recognition memory. Where do these associations come from? The architectural answer provided by Soar is that they must arise experientially by chunking (or else they are innate).

In fact, the NL-Soar model does not specify the associations that directly perform comprehension. It specifies a hierarchy of problem spaces that give rise to these associations via chunking. Figure 3.3 shows the basic structure (the details will be provided in the remainder of the chapter). When an impasse occurs due to lack of immediate knowledge to accomplish some comprehension operator function (proposal, selection, or application), NL-Soar enters these lower spaces where independent knowledge sources may be brought to bear in a search for the correct utterance or situation model. When the impasse is resolved, chunking automatically builds a new association in recognition memory that should allow comprehension to proceed smoothly in future similar situations. These associations may be quite general, or they may be quite specific, depending on the nature of the impasse and the problem solving. We call these associations *chunks*, though all associations have the same form, whether posited by the theorist or created by chunking.

One important characterization of comprehension that emerges from this model is that comprehension is a mix of recognition and deliberation. Given the severe time constraints, comprehension must be mostly recognition—an automatic, rapid process. But there is always the capability to fall back on the deliberate spaces when recognition fails. Just how much of adult comprehension consists of recognition vs. deliberation is an interesting theoretical and empirical issue that will be addressed in Chapter 7.

Another important characterization that emerges from this model is comprehension as a continuously improving skill. To be clear, NL-Soar does not specify a theory of language acquisition—the language-specific knowledge in the lower space is posited by the theorist. However, it does specify that certain aspects of comprehension will always be open to improvement. As we will see, the ability to handle difficult syntactic constructions, ambiguous material, and contextually specific interpretations may all be modulated by chunking. Language learning does not stop with the acquisition of syntax or vocabulary.

3.3 The utterance model

This section describes the structure of the utterance model and the processes for building it. The description is purely in syntactic terms, independent of semantics and context; the interaction of syntax with other knowledge sources will be explored in §3.6.

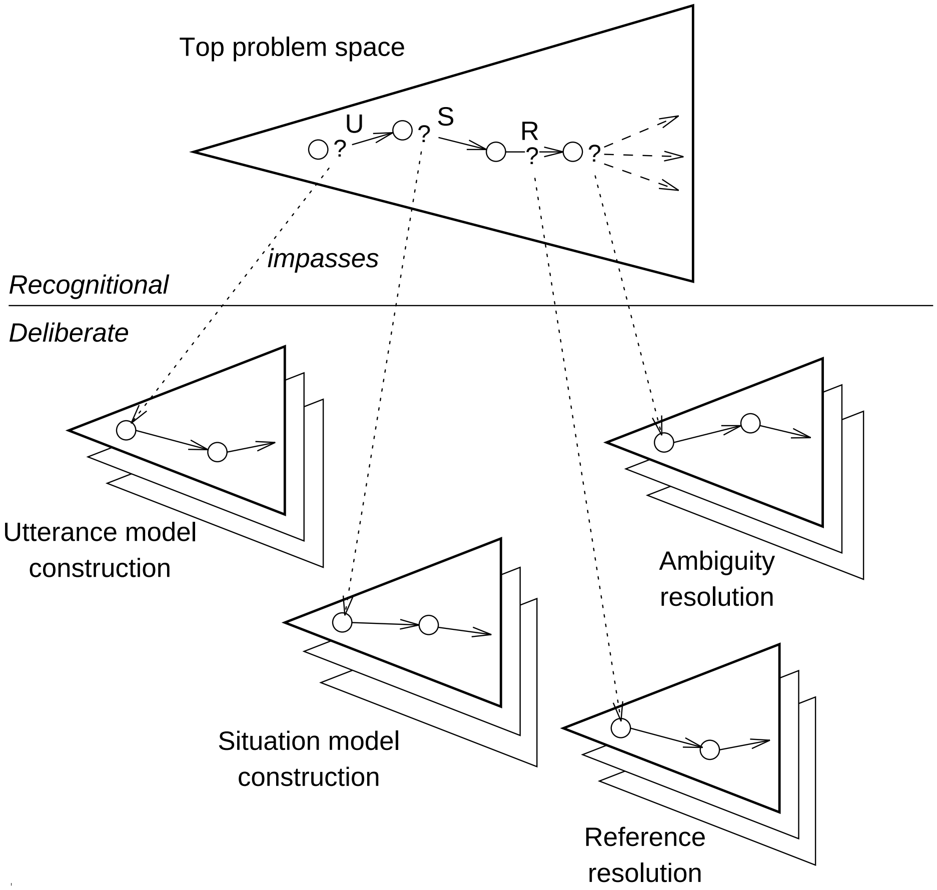


FIGURE 3.3: Comprehension as deliberation and recognition. The operators in the top space achieve comprehension by recognition when they are proposed, selected, and applied directly by immediate memory retrieval. If the required associations are not present in long-term memory, impasses arise and the relevant functions are carried out deliberately in lower spaces. As impasses are resolved, new chunks are formed that perform the function by recognition in future similar situations.

3.3.1 What the utterance model represents

The utterance model represents X-bar phrase structure as assumed in Government and Binding theory (e.g., (Chomsky, 1986; Cowper, 1992)). Because many of the predictions described later in the thesis are sensitive to syntactic structure, selecting an existing syntactic theory helps guard against ad hoc analyses that will fail to hold across a wider range of cross-linguistic structures. The particular choice of GB structures was made on both pragmatic and theoretical grounds. Pragmatically, using GB allows for a more direct comparison with the recent detailed models of Pritchett (1992) and Gibson (1991), as well as an incorporation of useful aspects of those models and analyses. Theoretically, the explicit principles and parameters approach fits naturally into the constraint-based generate-and-test framework of NL-Soar (described in the next section).

The basic X-bar schema is shown in Figure 3.4. X ranges over the syntactic categories

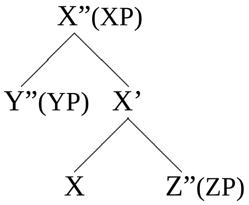


FIGURE 3.4: X-bar schema. YP is in specifier position (spec-XP). ZP is in complement position (comp-X’).

A (adjective), C (complementizer), I (inflection), N (noun), P (preposition), and V (verb). There are two levels of phrasal nodes projected from lexical heads: X’, and X”. X” is assumed to be the maximal projection and will usually be denoted XP. (Inflectional phrase (IP) corresponds to S in more traditional phrase structure grammars; complementizer phrase (CP) corresponds to S’.) The set of available syntactic relations between nodes is {spec, comp, comp2, head, adjoin, adjoin-head}, which denote the structural positions of specifiers, complements, heads, and adjunction. Adjunction will be explained in more detail below. Syntactic structure is thus a strict tree with a typical branching factor of one or two. Figure 3.5 gives the X-bar structure for a complex noun phrase , with the structural relations explicitly labeled. In future tree diagrams the relational labels will usually be omitted. Some intermediate nodes may also be dropped for brevity.

3.3.2 How the utterance model represents (or, why call it a *model*?)

We now make a general assumption about how mental models (§2.1.1) fit into Soar:

> (69) *Models assumption:* States in problem spaces are annotated mental models, which are pure models with a limited set of annotations or tags that expand the representational scope or help control processing (Newell, 1990).

This representational assumption is adopted in most cognitive modeling work in Soar. It grew out of Polk and Newell’s (1988) work in modeling syllogistic reasoning, which sought to explicate the role of mental models (ala Johnson-Laird) in Soar. As stated, it takes the form of an architectural assumption, since it cuts across all tasks and domains. However, as Newell (1990) points out, the attribute-value scheme in Soar is neutral with respect to the use of models or non-model representations. Thus, there is still an important issue as to what in Soar should give rise to this restriction. Without making any commitments to the genesis of models, we simply adopt (69) as a uniform representational law. The primary functional advantage of models is computational efficiency: the knowledge encoded in models can be extracted with match-like processing.

The immediate consequence of adopting this assumption with respect to syntactic structure in NL-Soar is that the representation of syntax must be a model. We call this

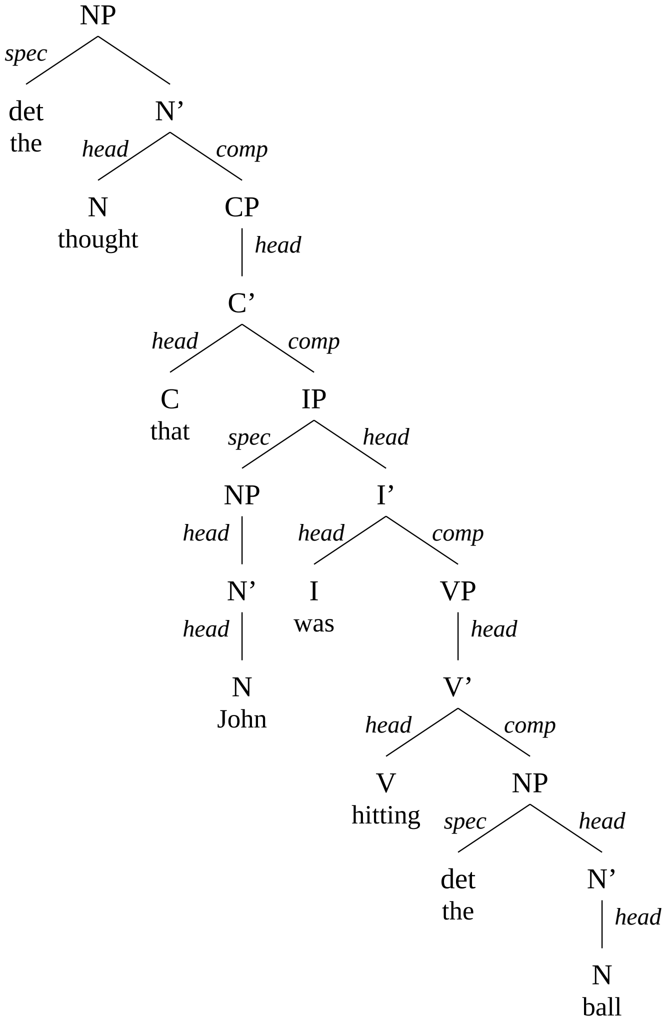


FIGURE 3.5: X-bar phrase structure for the complex NP *the thought that John was hitting the ball*.

representation an *utterance model* because it is a model of the structure of the utterance. The utterance model must satisfy the *structure correspondence principle* (§2.1.1), which states that aspects of a model correspond directly to aspects of the represented domain.

The realization of the utterance model as an attribute-value structure is straightforward: attributes correspond to the structural X-bar relations, or syntactic features such as category or agreement. The values of the attributes correspond to other objects in the model (i.e., nodes in the tree), or constants representing the values of the syntactic features. Figure 3.6 illustrates a simple example.

The model restriction may seem so weak as to provide little constraint in developing representations. But in fact, some familiar representations for syntax are ruled out because they violate structure correspondence. Pure logic representations are not models, as discussed earlier. The chart data structures that underlie the most efficient context free parsers (Earley, 1970; Winograd, 1983) also violate structure correspondence. Charts are space- and time-efficient because they systematically exploit redundancies across multiple

```
(p27 ^bar-level max    (n17 ^bar-level max
     ^head  p23)            ^head n45)

(p23 ^bar-level one    (n45 ^bar-level one
     ^head p19              ^head n6)
     ^comp n17)
                       (n6  ^bar-level zero
(p19 ^bar-level zero        ^category N
     ^category P            ^proper t
     ^lex for)              ^lex Allen)
```

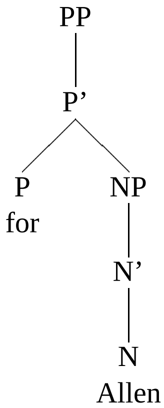


FIGURE 3.6: Attribute-value structures for the utterance model. Each node in the utterance model is represented by a unique identifier in working memory, with a set of feature augmentations (attributes and values).

structural interpretations. As a result, the correspondence between elements in the chart and elements in the represented domain becomes one-to-many. The efficiency of constructing and storing the chart is traded off against the potentially increased computation required to extract information from the chart; for example, determining whether a chart represents a particular phrase structure tree involves a combinatoric search through the possible configurations implicit in the chart.

3.3.3 Constructing the utterance model

This section describes the processes that incrementally build the utterance model. First we consider how the utterance model is organized in working memory, then trace the construction of the model from lexical access to establishing syntactic relations.

The utterance model in working memory

As illustrated in Figure 3.6, the utterance model is an attribute value structure. The structure must be anchored in some fashion to the problem space state, which is a distinguished object in the goal context in Soar's working memory (this is an architectural requirement). All objects in a state are attached to the state identifier as values of attributes. For the utterance model, one possibility is to simply have a single attribute which points to the root of the phrase structure tree:

```
(state9 ^root u39)
(u39 ^bar-level max ^head ...)
```

Of course, multiple values will be required when the utterance model consists of multiple constituents not yet grouped into higher structures:

```
(state9 ^root u39 u40)
(u39 ^bar-level max ...)
(u40 ^bar-level zero ..)
```

The attributes at the state level define the initial access path to the utterance model. With a root attribute, the root node is directly accessible; any other node must be reached via the root. This requires explicitly encoding the specific access paths into the conditions of associations in recognition memory, and/or providing a general problem space for navigating the structures.

Another possibility is to provide uniform access to all nodes in the tree:

```
(state9 ^all-nodes u39 u40 u42 u57 ...)
```

These two possibilities define two extreme points of a range of possible indexing schemes. There is a basic computational tradeoff between the cost of the recognition memory match, and the cost of the deliberate problem solving. The root-access scheme requires encoding specific access paths in chunks, which means there is increased potential for deliberate problem solving when the set of existing chunks is insufficient for accessing some novel structure. The uniform access scheme avoids the necessity of encoding specific access paths. However, the uniform access path involves a large multiply-valued attribute (or *multi-attribute*) which gives rise to unwanted combinatorics in the match. The potential combinatorics caused by these undiscriminated sets can easily be seen in the following condition for an association. Angle brackets denote variables:

```
(<state> ^all-nodes <x> <y>)
```

Given *n* values on the `all-nodes` attribute, this particular condition will lead to n^2 potential instantiations of the association. In general, with *C* conditions and *W* working memory elements, the match complexity is W^C. An association with such a condition is called an *expensive chunk*, and the exponential cost of matching these associations has been demonstrated repeatedly both in formal analysis and in implemented Soar systems (Tambe, Newell & Rosenbloom, 1990). The resulting slowdown compromises the constant time assumption of the match (Table 3.1).

To avoid the pitfalls of both extremes, NL-Soar uses an accessing structure that provides sufficient discrimination to avoid the expensive chunk problem, yet is functionally adapted to the parsing process in such a way that the relevant nodes in the utterance model are directly accessible.

The idea is to index nodes by the potential syntactic relations that they may enter into. The structure is called the *A/R set*, for assigners and receivers. Figure 3.7 shows the contents of the A/R set during parsing *John hit the ball*, just after the NP [$_{NP}$ *the ball*] has been formed, and just before attaching [$_{NP}$ *the ball*] in the complement position of [$_{V'}$ *hit*]. The NP [$_{NP}$ *the ball*] is a potential receiver of the relation *complement of V'* (object of a verb), as well as *specifier of IP* (subject position). The V' node projected from *hit* is a potential assigner of the complement of V' relation.

ASSIGNERS	adjoin-N’:	[$_{N'}$ *ball*]
	comp-V’:	[$_{V'}$ *hit*]
	adjoin-V’:	[$_{V'}$ *hit*]
RECEIVERS	comp-V’:	[$_{NP}$ *the ball*],[$_{CP}$ *John hit*]
	adjoin-IP:	[$_{CP}$ *John hit*]
	spec-IP:	[$_{NP}$ *the ball*]

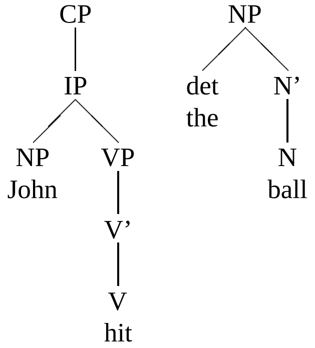


FIGURE 3.7: The A/R set during *John hit the ball.* [$_{N'}$ *ball*] can assign an adjoin-N’ (modifier) relation, [$_{NP}$ *the ball*] can receive the comp-V’ relation (be a complement of a verb), and so on.

Proposing potential links in the utterance model is a matter of pairing up assigners and receivers indexed by the same relation. For example, the following condition binds to a verb and a potential object:

```
(<state> ^assigners-comp-V' <v1>
         ^receivers-comp-V' <xp>)
```

Once a link is made, the receiving node is removed from the receivers set for all relations, since a node can only have one parent in the tree. It also seems reasonable to remove the assigners node from the assigners set for the particular relation established. However, this can lead to disaster. The A/R set provides general access to the utterance model not only for parsing, but also for interpretation. If nodes were removed immediately from the assigners set when links were established, the node could potentially disappear entirely from working memory before any interpretation could take place. Thus, the nodes remain in the assigners set even after links are established (Figure 3.7 shows just a subset of the complete A/R set). The assigners set, then, provides an efficient access mechanism for any process that works on the partially completed syntactic structures. For example, the following condition instantiates for a verb and its object:

```
(<state> ^assigners-comp-V' <v1>)
(<v1> ^comp <obj>)
```

Eliminating open undiscriminated sets in working memory

Although the A/R set helps reduce the size of undiscriminated sets in working memory, it does not completely avoid them, since even specific attributes like `assigners-comp-V'` can grow indefinitely. One way to eliminate the problem completely is to restrict all attributes to single values. This is the *uni-attribute* approach (Tambe et al., 1990). NL-Soar adopts a similar approach, but fixes the set of possible values to *two.* The motivation

ASSIGNERS	adjoin-A:	[$_A$ *square*]
	adjoin-N:	[$_N$ *square*]
RECEIVERS	head-A’:	[$_A$ *square*]
	head-N’:	[$_N$ *square*]

A	N
square	square

FIGURE 3.8: Results of lexical access. Each potential bar-level zero node is retrieved and placed in the A/R set.

for the particular value two is discussed in Chapter 5. For now it is enough to realize that the restriction helps to avoid the unbounded growth of syntactic structures in working memory, and that this method follows from the empirical and theoretical investigation of the recognition match in Soar.

Lexical access

Lexical access provides the raw material for constructing the utterance model. More specifically, lexical access retrieves a set of nodes corresponding to all the bar-level zero positions that the lexical item may occupy. These nodes are deposited into the appropriate locations in the A/R set. Figure 3.8 shows the results of the lexical access of *square*. Two nodes are retrieved corresponding to the adjectival and nominal senses of the word. The nodes are indexed in the receivers set under `A’-head` and `N’-head`. In other words, *square* can serve as the head of an adjective phrase or a noun phrase.

Lexical access is accomplished by associations in recognition memory, like any other process in Soar. The access of the multiple entries happens in parallel and independently of context. The parallelism is due strictly to the inherent parallelism of the recognition match. The context independence is an assertion about the conditions of the access associations. There is a fair amount of empirical evidence that suggests lexical access is independent of biasing contexts (e.g., (Swinney, 1979)). However, there are also some functional reasons for assuming context independence, discussed in §3.7.

NL-Soar does not provide a detailed model of lexical access. It simply circumscribes the required functionality by positing associations that map words to their entries as described above. Given these mapping associations, there are at least two distinct possibilities for realizing lexical access: by operator application or by encoding associations. If the access happens via operator application, then the associations will be conditional upon some particular operator (perhaps a special lexical-access operator). If the access happens via encoding, then the associations can fire independently of the current goal context and focus of attention. There is some reason to believe that the latter may be correct; the Lackner & Garrett (1972) experiments suggest that lexical access can occur without attention. Since encoding productions are not yet implemented in Soar, NL-Soar ties the access to an operator application. This choice is not critical to most of the predictions made in the subsequent chapters. When there is any effect, it will be explicitly noted.

Head-driven, constraint-based construction

Once lexical access is accomplished, the construction of the utterance model proceeds by establishing structural relations between nodes. Figure 3.9 shows the processing after encountering *Florida* in *John likes Florida.* The relations are created by *link* operators. These link operators exist in one of the lower problem spaces that implement the u-constructors. As Figure 3.9 illustrates, some of the link operators are used to create the higher projections of the level zero nodes retrieved from lexical access.

Each link may be subject to a number of syntactic constraints. These constraints are independently represented by separate operators in a problem space that checks the well-formedness of proposed links. For example, Figure 3.10 shows the constraints evoked for the link operator that places [$_{NP}$ *John*] in specifier (subject) position. The agreement constraint ensures that the subject and verb agree in number. The order constraint ensures that the subject precedes the verb. These constraints are generated for any X-bar specifier, including determiners for noun phrases. NL-Soar currently implements a number of basic constraints, including number/person agreement, order, subcategorization, and simple case checks.

This parsing organization has two distinguishing features:

- Parsing is a *bottom-up, head-driven* process that begins with projecting phrasal nodes from their incoming lexical heads (Pritchett, 1991). Nodes are only created when there is lexical evidence for them. In other words, there are no explicit expectation data structures.

- There are no explicit phrase structure grammar rules in the system. Instead, the utterance model emerges from the interaction of the X-bar structures projected from lexical heads, and the independently represented constraints.

NL-Soar's parsing mechanisms thus naturally reflect the basic structure of grammar assumed in the principles and parameters approach to syntax. While the constraints implemented in the present model do not necessarily map directly onto the principles or modules in GB, a closer and more thorough mapping should be possible. In any event, the structure already present in NL-Soar is consistent with a lexically-generated, constraint-based approach to grammar.

Besides the simplicity of the mechanisms and the foundation in linguistic theory, there are good functional reasons for adopting the present scheme. Bottom-up processing is well-suited for handling fragmentary input. Head-driven parsing does not encounter the spurious ambiguity inherent in more top-down approaches that must select among various phrase structure rules that predict incoming structure (Abney, 1989). The implications of using head-driven parsing with head-final languages will be dealt with in Chapter 8.

Adjunction

The examples above focus on projections, or attaching structures in specifier or complement position. The other means of joining two structures is *adjunction*, which is used for all

ASSIGNERS	adjoin-V’:	[$_{V'}$ *likes*]
	comp-V’:	[$_{V'}$ *likes*]
RECEIVERS	comp-V’:	[$_{CP}$ *John likes*]
	head-N’:	[$_{N}$ *Florida*]

LINK

ASSIGNERS	adjoin-V’:	[$_{V'}$ *likes*]
	comp-V’:	[$_{V'}$ *likes*]
RECEIVERS	comp-V’:	[$_{CP}$ *John likes*]
	head-NP:	[$_{N'}$ [$_{N}$ *Florida*]]

LINK

ASSIGNERS	adjoin-V’:	[$_{V'}$ *likes*]
	comp-V’:	[$_{V'}$ *likes*]
RECEIVERS	comp-V’:	[$_{NP}$ [$_{N'}$ [$_{N}$ *Florida*]]], [$_{CP}$ *John likes*]
	spec-IP:	[$_{NP}$ [$_{N'}$ [$_{N}$ *Florida*]]]

LINK

ASSIGNERS	adjoin-V’:	[$_{V'}$ *likes*]
	comp-V’:	[$_{V'}$ *likes*]
RECEIVERS	comp-V’:	[$_{CP}$ *John likes*]

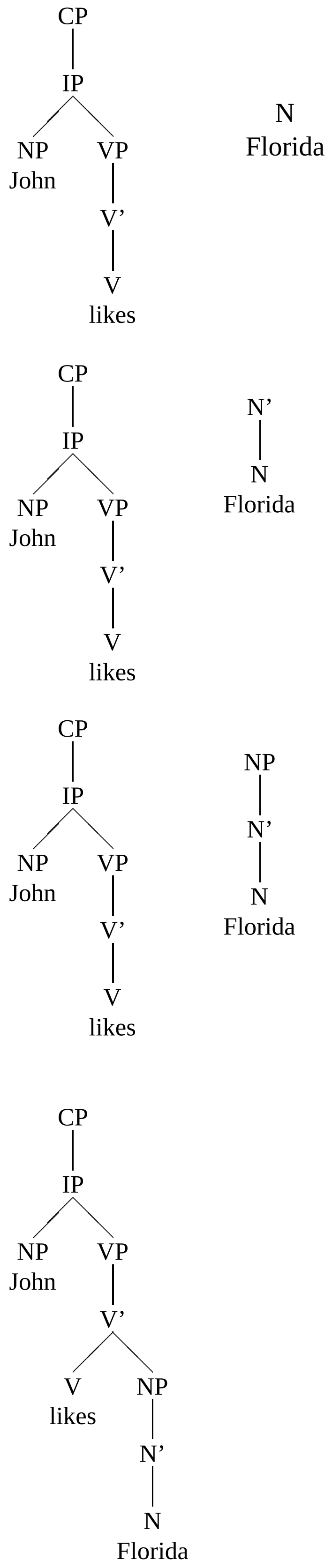


FIGURE 3.9: Building the utterance model with link operators. *Florida* is first projected to an NP with two links, then attached as the complement of *likes*.

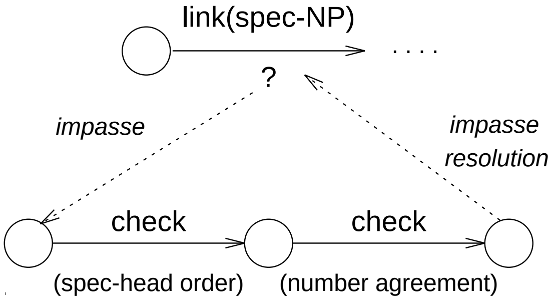


FIGURE 3.10: Checking constraints for proposed links. Each syntactic link may evoke a number of independent constraints.

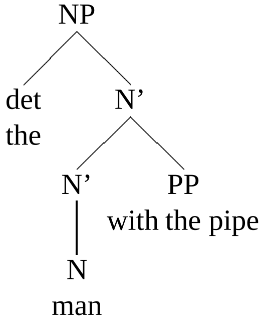


FIGURE 3.11: An adjoined structure. Adjunction introduces a new node in the tree (in this case, a new N' node.)

non-argument (i.e., non-positionally-marked relations) modification. Adjuncts are assumed to be Chomsky-adjoined (Chomsky, 1986), meaning adjunction results in the creation of an additional node. Figure 3.11 gives an example of an adjoined structure. The relations *adjoin* and *adjoin-head* are used to distinguish the structure from the basic specifier, complement, and head structures. Adjunction is assumed to be uniform in that any phrasal level (zero, one, maximal) may be adjoined to (cf. (Chomsky, 1986)). Like all structure building, adjunction is realized in NL-Soar by link operators. When the link operators establishes an adjoin relation, they simply create the additional node.

Traces and long-distance dependencies

Deep structure relations are represented by phonologically null trace elements in the syntax trees (Chomsky, 1981). Consider a simple wh-question, in which the wh-pronoun is related to the missing object of the verb:

(70) Who$_i$ did Orlando draft t$_i$?

The relationship is represented by *coindexing* the pronoun and the trace element in the complement position, which establishes their referential and syntactic equivalence. Figure 3.12 shows the phrase structure for (70).

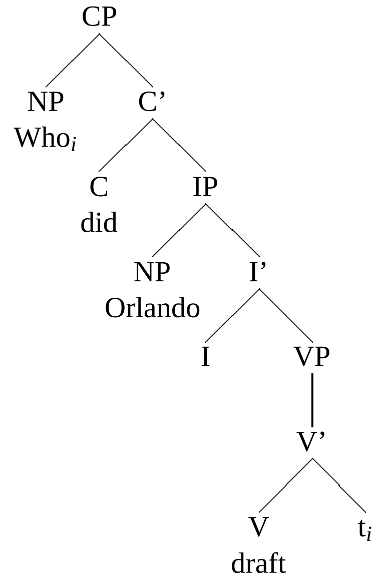


FIGURE 3.12: Traces in syntactic structure.

NL-Soar generates trace elements as it does any structural relation: with the link operator. The coindexation is handled in the current system by establishing a pointer to the antecedent, though it could be handled by copying the relevant features from the trace's antecedent onto the newly established node.

The proposal of link operators that create traces is triggered by the presence of potential antecedents. There is no need to wait for the "surface position" of the trace, contrary to (Pickering & Barry, 1991). In particular, these proposals test the spec-CP position. The proposal conditions for traces in verb complement position look like:

```
(<state> ^assigners-spec-CP <cp>
         ^assigners-comp-V' <v1>)
(<cp> ^spec <np>
      ^head.comp.head.comp.head <v1>)
```

The attribute name with the dot notation is simply a shorthand for following along a path of multiple structural links.

This mechanism handles arbitrarily long distance dependencies as well:

(71) Who$_i$ do you think the media believes Orlando will draft t$_i$?

Such dependencies can be handled because the syntactic structure assigned by GB (Figure 3.13) breaks down the long-distance relationship into a chain of short links between intermediate trace elements. A universal locality constraint (subjacency) ensures that each link in the chain is established over a local section of the tree. Thus, the fixed local patterns in the proposals for the link operator suffice to create these chains.

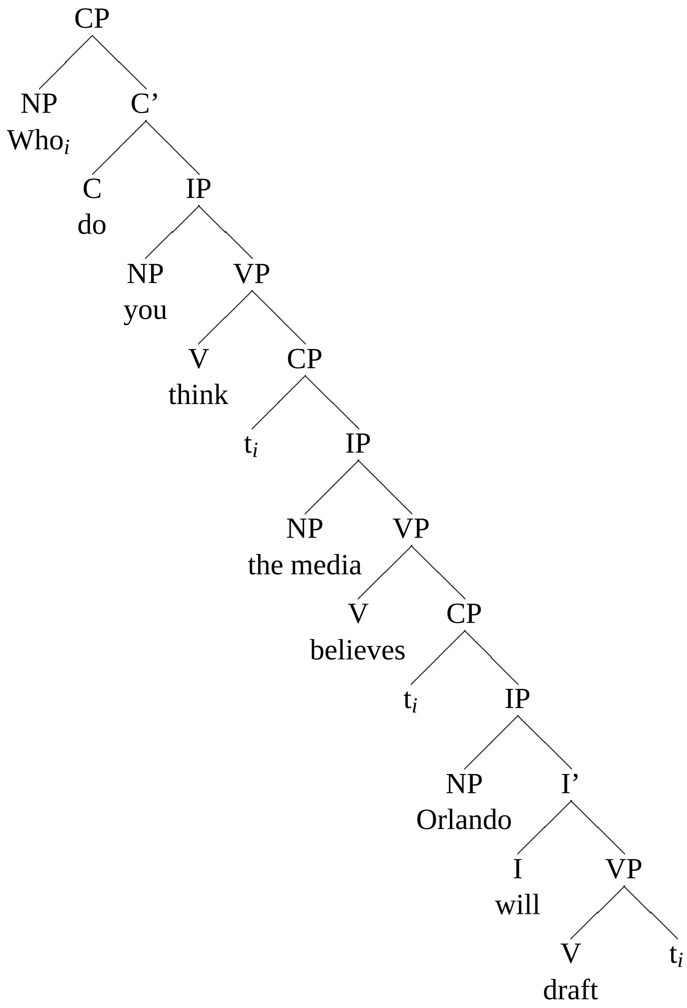


FIGURE 3.13: Phrase structure for a long distance dependency. The relationship between *who* and the object of *draft* is established by a chain of local relations.

Single path and destructive repair

NL-Soar is a single path comprehender. This is not an additional assumption, but one that follows naturally from the models assumption (69) and a basic architectural feature of Soar (the single state principle). The derivation goes as follows:

1. Problem space states are annotated models (the models assumption).

2. Models represent one situation (a basic property of models).

3. Problem space states represent one situation (from 1 and 2).

4. Soar maintains a single state per active problem space; previous states are not available to backup to (the single state principle (Newell, 1990)).

5. Therefore, NL-Soar is a single path comprehender (from 3 and 4).

Actually, the argument is not quite as tight as this. There are two ways of slipping in some characteristics of a multi-path comprehender, so that the claim is weakened somewhat[7]. First, the identification of problem space states as annotated models may be given a weaker interpretation by allowing single states to contain more than one model[8], thereby representing more than one situation. In fact, we will see below how NL-Soar permits some momentary limited parallelism in violation of the strict one pure model assumption. Of course, one of the basic findings of research into the nature of mental models is that it is very difficult to manipulate multiple models (Johnson-Laird, 1983; Johnson-Laird, 1988).

Second, the annotations in annotated models violate structure correspondence by definition and may therefore permit representations of multiple situations with a single model. For example, one annotation explored in the work on syllogistic reasoning (Polk, 1992) is *optional,* which means that the annotated object may or may not be in the represented situation. However, the semantics of annotations are constrained such that they must be interpreted with respect to a local piece of the model. Arbitrary scoping domains cannot be established, which would begin to approach the power of first order logic.

Even granting these possible exceptions, it seems clear that uniformly adopting the models assumption in Soar leads to a comprehension theory that tends strongly to a single path model. Interestingly, the restriction to a single model per state is equivalent to eliminating *arbitrary copy on demand*, at least in the case of the utterance model. If multiple syntactic interpretations are permitted, then at an *n*-way branching point (local ambiguity), *n* complete copies of the existing model(s) must be produced to generate the new independent models. (As noted earlier, structure sharing violates structure correspondence). Thus, the single path assumption can also be seen as a way of dispensing with an arbitrary copy mechanism.

The critical functional question that the single path assumption gives rise to is: What happens when the incoming input is inconsistent with the chosen structure?

Consider the case of local lexical ambiguity in (72):

(72) The square table is large.

The parallelism of the lexical access leads to a momentary parallelism of syntactic structure which violates the strict single model assumption. Figure 3.14 shows what happens, tracing the contents of the A/R set and showing the evolving phrase structure. When *square*

[7] There is one other possibility in addition to these two. Although Soar maintains only one state per problem space, the goal stack may grow indefinitely, and each new problem space context can be used to maintain a unique state. There are several problems with this approach that make it rather implausible. It could only be used to store states for backtracking, rather than advancing multiple states in parallel. The reason is simple: each time an operator is applied to one of the states, the entire goal context below that state is destroyed and garbage collected. Furthermore, the amount of impassing that is required to generate the context stack introduces a processing overhead that makes it highly unlikely that real-time comprehension could be achieved.

[8] The utterance and situation models do not count as multiple models here, since they are different *types* of models representing different aspects of the same situation. The important thing is that multiple possible situations are not being represented. In this view, it may be more accurate to refer to the utterance and situation models as *submodels*, but I will stay with the standard terminology.

arrives, NP and AP nodes are projected, and the determiner *the* is attached in spec-NP position, forming the NP [$_{NP}$ *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.

Such local inconsistencies cannot be allowed to persist and propagate. They would eventually lead to complete functional breakdown because the model representation does not systematically support multiple interpretations. To repair the structure, NL-Soar has an additional operator, *snip*, which breaks a link previously established by a link operator.

Figure 3.15 shows how snip works to repair the structure in Figure 3.14. The snip operator is immediately triggered by the presence of syntactic structure attached to competing senses of the same lexical token. Preference is given to the more recent structure, so snip breaks the spec 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*].

Snip is the minimal amount of new mechanism required to effect repairs. In fact, snip does not complete the repair, it just destroys a bit of structure and then lets the link operators take over to finish the job. Snip is in the class of repair mechanisms called *simple destructive repair* (Lewis, 1992). It works on the existing structure, without recourse to any previously held states, and the repair itself is accomplished by the existing constructors.

Consider another kind of momentary inconsistency that does not arise from lexical ambiguity:

(73) John knows Shaq is tall.

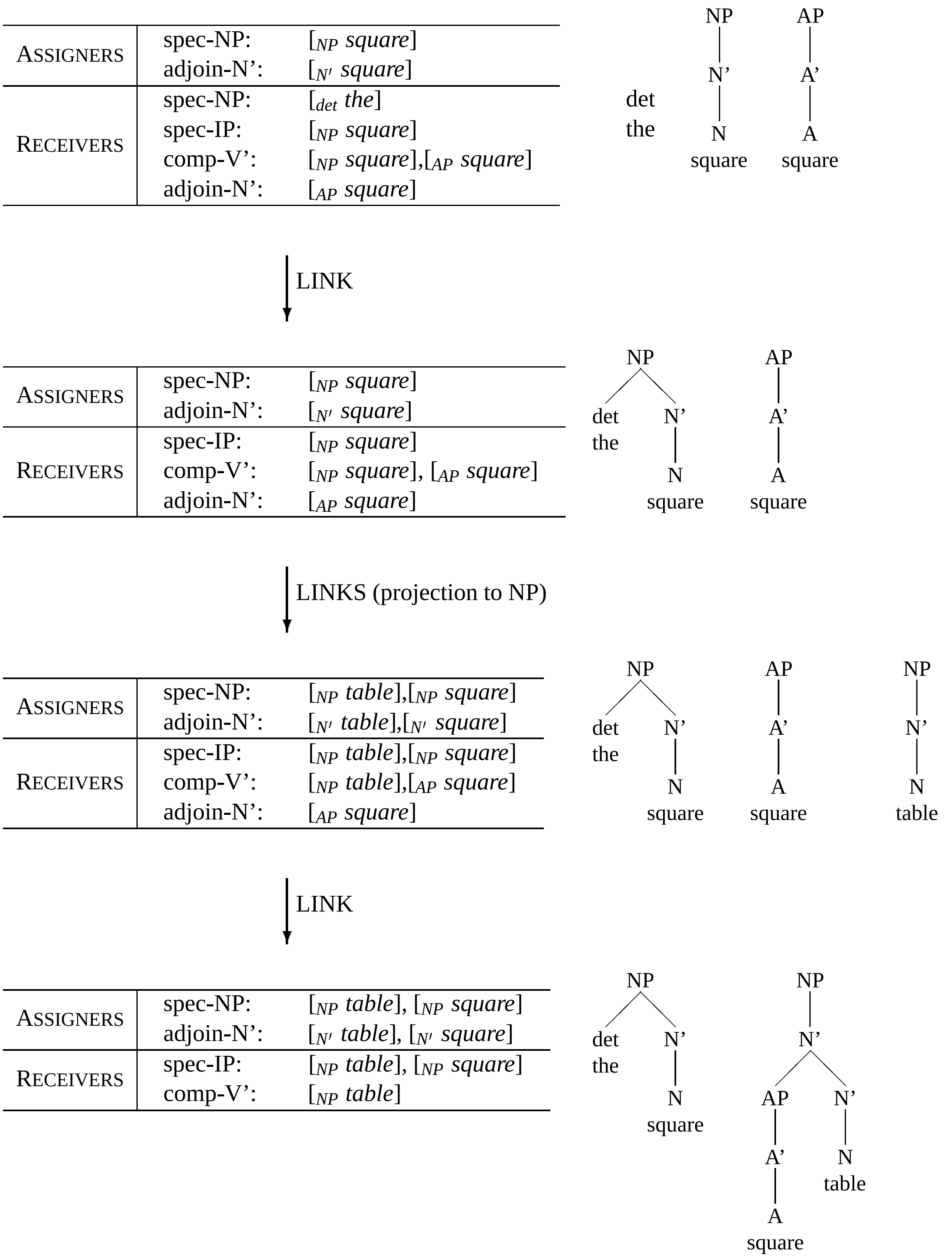
Figure 3.16 shows what happens. *Shaq* is initially attached in the complement position of *knows*. When *is* arrives, it is projected to an IP and CP (CPs are projected in the absence of overt complementizers, following Pritchett (1992)). Next, a link is proposed to attach the CP in the complement position of *knows*. This proposal is made because *knows* is still on the assigners set. The proposed link is well-formed since *knows* can take sentential complements as well as nominals.

The result, as in the case of the lexical ambiguity, is a momentary parallelism of structure. The utterance model is in effect a superposition of two separate structures, with two phrase markers competing for the same structural position (comp-V' of *knows*). This local inconsistency triggers the 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 generation of snip as described is highly constrained. Snip is proposed only in the following cases:

1. When incompatible projections of the same lexical token are both attached to other lexical structure (e.g., the case of *square* in (72));

2. When an inconsistency is detected local to a particular maximal projection (e.g., the case of *knows* in (73)).



ASSIGNERS	spec-NP:	[$_{NP}$ *square*]
	adjoin-N’:	[$_{N'}$ *square*]
RECEIVERS	spec-NP:	[$_{det}$ *the*]
	spec-IP:	[$_{NP}$ *square*]
	comp-V’:	[$_{NP}$ *square*],[$_{AP}$ *square*]
	adjoin-N’:	[$_{AP}$ *square*]

ASSIGNERS	spec-NP:	[$_{NP}$ *square*]
	adjoin-N’:	[$_{N'}$ *square*]
RECEIVERS	spec-IP:	[$_{NP}$ *square*]
	comp-V’:	[$_{NP}$ *square*], [$_{AP}$ *square*]
	adjoin-N’:	[$_{AP}$ *square*]

ASSIGNERS	spec-NP:	[$_{NP}$ *table*],[$_{NP}$ *square*]
	adjoin-N’:	[$_{N'}$ *table*],[$_{N'}$ *square*]
RECEIVERS	spec-IP:	[$_{NP}$ *table*],[$_{NP}$ *square*]
	comp-V’:	[$_{NP}$ *table*],[$_{AP}$ *square*]
	adjoin-N’:	[$_{AP}$ *square*]

ASSIGNERS	spec-NP:	[$_{NP}$ *table*], [$_{NP}$ *square*]
	adjoin-N’:	[$_{N'}$ *table*], [$_{N'}$ *square*]
RECEIVERS	spec-IP:	[$_{NP}$ *table*], [$_{NP}$ *square*]
	comp-V’:	[$_{NP}$ *table*]

FIGURE 3.14: How structural parallelism arises in the utterance model. Lexical access retrieves multiple senses of the same word in parallel, which can potentially lead to the simultaneous incorporation of competing syntactic senses. Here, *square* is attached as both a noun and a verb—a momentary inconsistency.

ASSIGNERS	spec-NP: adjoin-N’:	[$_{NP}$ *table*], [$_{NP}$ *square*] [$_{N'}$ *table*], [$_{N'}$ *square*]
RECEIVERS	spec-IP: comp-V’:	[$_{NP}$ *table*], [$_{NP}$ *square*] [$_{NP}$ *table*]

SNIP

ASSIGNERS	spec-NP: adjoin-N’:	[$_{NP}$ *table*], [$_{NP}$ *square*] [$_{N'}$ *table*], [$_{N'}$ *square*]
RECEIVERS	spec-NP: spec-IP: comp-V’:	[$_{det}$ *the*] [$_{NP}$ *table*], [$_{NP}$ *square*] [$_{NP}$ *table*]

LINK

ASSIGNERS	spec-NP: adjoin-N’:	[$_{NP}$ *table*], [$_{NP}$ *square*] [$_{N'}$ *table*], [$_{N'}$ *square*]
RECEIVERS	spec-IP: comp-V’:	[$_{NP}$ *table*], [$_{NP}$ *square*] [$_{NP}$ *table*]

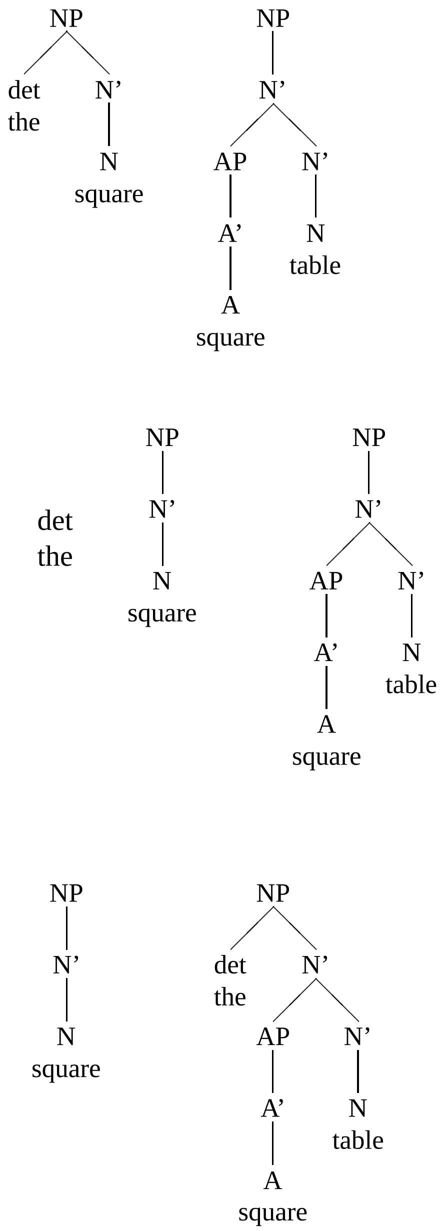


FIGURE 3.15: Repairing an inconsistency with the snip operator. The incorporation of competing syntactic senses triggers a snip operator to detach one of the senses. Here, the determiner is snipped from the noun sense of *square*, making it available to attach to the globally correct noun phrase, *the square table*.

Assigners	spec-IP:	[$_{IP}$ *is*]
	adjoin-V’:	[$_{V'}$ *knows*]
	comp-V’:	[$_{V'}$ *knows*]
Receivers	comp-V’:	[$_{CP}$ *is*], [$_{CP}$ *John knows*]

LINK

Assigners	spec-IP:	[$_{IP}$ *is*]
	adjoin-V’:	[$_{V'}$ *knows*]
	comp-V’:	[$_{V'}$ *knows*]
Receivers	comp-V’:	[$_{CP}$ *John knows*]

SNIP

Assigners	spec-IP:	[$_{IP}$ *is*]
	adjoin-V’:	[$_{V'}$ *knows*]
	comp-V’:	[$_{V'}$ *knows*]
Receivers	comp-V’:	[$_{NP}$ *Shaq*], [$_{CP}$ *John knows*]
	spec-IP:	[$_{NP}$ *Shaq*]

LINK

Assigners	spec-IP:	[$_{IP}$ *is*]
	adjoin-V’:	[$_{V'}$ *knows*]
	comp-V’:	[$_{V'}$ *knows*]
Receivers	comp-V’:	[$_{CP}$ *John knows*]

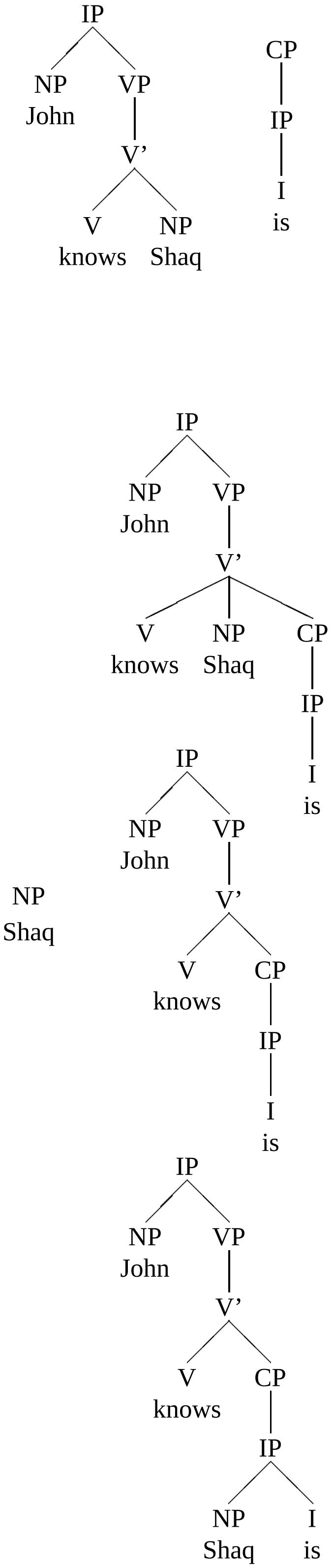


Figure 3.16: Repairing a complement inconsistency with snip. The incoming CP vies for the same complement position as the NP (*Shaq*), triggering a snip to detach *Shaq*. Next, *Shaq* is attached as the subject of the incoming clause.

There are good computational reasons for such a tightly constrained generation. A free generation of snip for every link in the utterance model has two undesirable consequences. First, it leads directly to a potentially large, undiscriminated set of operators in working memory. As discussed earlier, such sets (multi-attributes) are a source of exponential match cost in the recognition memory. Second, even for moderately-sized syntactic structures, the introduction of freely generated snips increases the search space significantly.

Thus, this constrained repair mechanism avoids significant increases in both knowledge search (recognition match) and problem search (growth of problem space). However, it provides the functionality required to deal with unproblematic ambiguities; Chapter 6 explores the mechanism in detail on the 57-sentence collection of unproblematic ambiguities and garden path sentences.

3.3.4 Chunking new u-constructors

The processes described above for constructing the utterance model are organized into a problem space hierarchy that produces new u-constructor comprehension operators. The complete hierarchy is shown in Figure 3.17. We will step through this Figure, showing how one particular u-constructor is constructed while parsing the sentence *John knows Shaq is tall*. We trace the processing after *Shaq* has been comprehended, and the lexical access for *is* has just completed.

Learning a brand new operator in Soar requires learning two things at minimum: the proposal of the new operator, and its implementation. The impasse (1) in the *top* space of Figure 3.17 indicates that no operators are available to continue comprehension. This impasse provides the opportunity to learn the proposal associations for a new operator.

Soar responds to the impasse by dropping into the *Create-operator* space, which contains the capability to create new symbols (gensyms) designating new operators. The arbitrary symbol `u-constructor17` is created, and an operator with that name is proposed and selected. Of course, another impasse (2) immediately occurs since it is not yet known what this operator will actually do. This is an operator implementation impasse, and provides the opportunity to learn the implementation associations for the new operator. The first impasse (1) remains unresolved, since the operator has not been proposed in the higher space.

Soar responds to this impasse by dropping into the *U-construct* space, which contains the primitive operators (link and snip) that build the utterance model. The state is shared with the Create-operator space. A series of link operators (3) fire that project the verb *is* through IP and CP nodes. As the structure is built, new associations are automatically created through chunking. These associations implement the new operator. For example, one of the chunks has the form:

IF the operator is `u-constructor17`
and there is a node X in the receiver's set indexed by head-IP
THEN create an IP node with X as its head

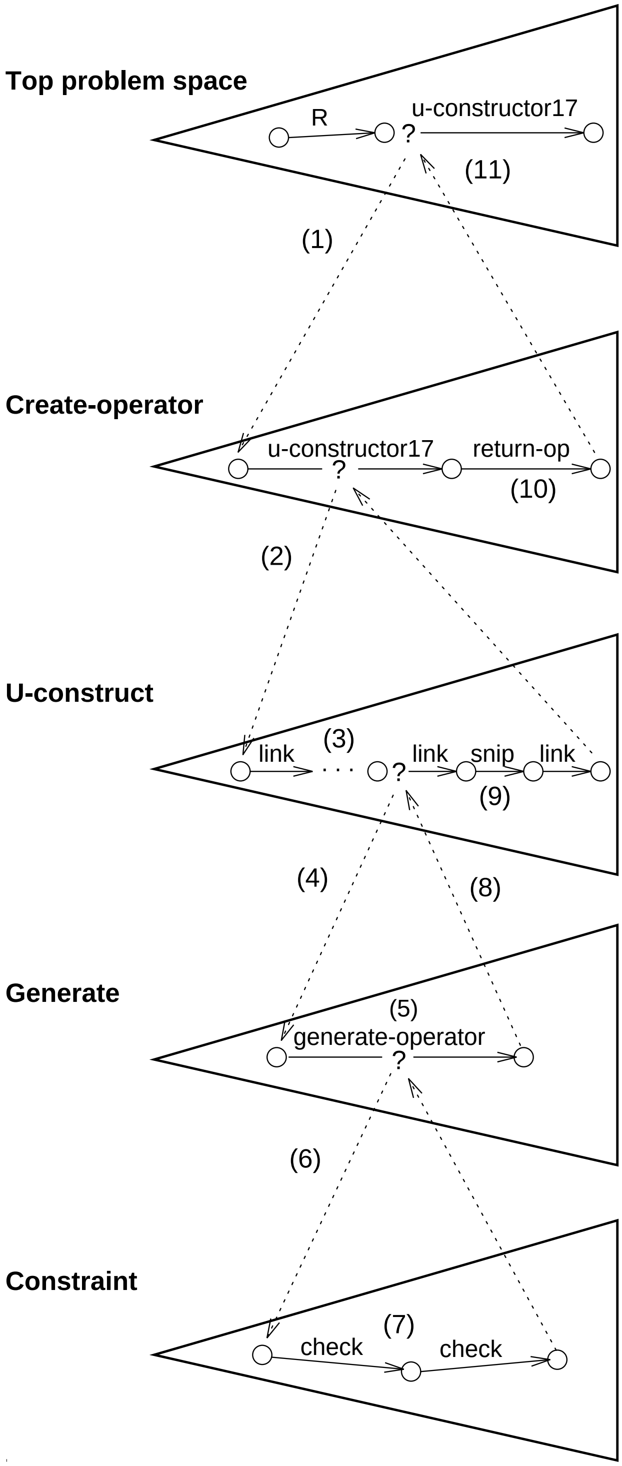


FIGURE 3.17: The problem spaces for building u-constructors. *Create-operator* generates new comprehension operator names; in this case, `u-constructor17`. *U-construct* assembles the utterance model via primitive structure building operators. A sequence of primitive operators becomes a single u-constructor. Syntactic well-formedness constraints are checked in the *Constraint* space.

Another impasse arises (4) due to the unavailability of additional link operators. This provides the opportunity to learn proposals for new link operators. Soar then enters the *Generate* space, which contains the basic generator for links. The *generate-operator* operator matches potential assigners and receivers that are adjacent; thus, the proposal conditions for this operator embody X-bar structure. In this case, generate-operator (5) proposes a complement link between *knows* on the assigner set, and the newly projected CP on the receiver set. A generated operator is returned to the U-construct space only if it is well-formed according to all the syntactic constraints. Thus, another impasse arises (6) because it is not immediately known whether the well-formedness constraints for the complement link are satisfied.

Finally, Soar enters the bottom space in the hierarchy, the *Constraint* space, where the independent syntactic constraints are proposed and checked (7). In this case, the proposed complement link must satisfy a subcategorization check (to make sure this verb can take a CP) and an order check (to make sure the complement follows the head). These checks pass, allowing the generate-operator to complete, which in turns allows the link operator to be proposed in the U-construct space (8). This causes impasse (4) to be resolved, building a chunk which specifically proposes CP complement links. The conditions of the chunk integrate the independent constraints checked in the Constraint space:

IF a node X is in the assigners set indexed by comp-V'
and X can take a CP complement
and node Y is in the receivers set indexed by comp-V'
and Y is a maximal projection of category C
and Y follows X and is adjacent to X
THEN propose the link operator for a complement link between X and Y

The complement link is performed, producing more implementation chunks for the new operator `u-constructor17`. When the link is complete, a snip operator is immediately proposed (9), triggered by the two structures competing for the same position as described earlier. The snip breaks the complement link between *knows* and *Shaq*, producing additional implementation chunks for `u-constructor17`. The resulting state satisfies the conditions for the link operator that establishes NPs in subject (spec-IP) position. This link completes the implementation of `u-constructor17`.

The successful completion of the u-constructor application triggers the *return-operator* operator (10) in the Create-operator space. Return-operator simply proposes `u-constructor17` in the top space (11). The resulting chunk has the form:

IF there is a node X in the receivers set indexed by head-I'
and a node Y on the assigners set indexed by comp-V'
and Y assigns a complement role to an NP Z
and Y can take a CP complement
and Z and X agree in number and X follows and is adjacent to Y and Z
THEN propose `u-constructor17`

The conditions of the proposal chunks integrate over all the various constraints required by the links and snips that ultimately comprised the implementation of the u-constructor. The chunk is not specific to particular words, since any lexically specific information required for building the utterance model is retrieved by lexical access prior to the impasses. After chunking, the behavior looks like Figure 3.2. The proposal chunk for `u-constructor17` fires, the operator is selected, and the implementation chunks directly apply the operator without impasse.

General features of u-constructors

`U-constructor17` illustrates a number of important general features about u-constructors and their creation in NL-Soar:

- A single u-constructor may be composed of many primitive structure-building operations. The u-constructor collapses into one operator application the sequential problem solving that involves generating and testing potential links against independent constraints.

- There is no fixed vocabulary or number of u-constructor types. Any sort of operator may be produced, ranging from fairly simple constructions to relatively complex and specific constructions, as `u-constructor17` demonstrates. `U-constructor17` may be interpreted as the "project incoming verb to complement phrase, attach as sentential complement and reanalyse the previous NP complement as the subject of the sentential complement" operator. The set of available u-constructors defines the recognitionally available syntactic knowledge in the system.

- The content of the operator is not specified by explicit data structures which must be interpreted to yield the behavior. All that appears in the top space is the gensymed name of the operator. What that symbol refers to is held in the implementation associations in recognition memory. The u-constructors may correspond to specific syntactic constructions, but they cannot be viewed as explicit phrase structure rules.

- The repair process is seamlessly integrated as a part of recognitional comprehension at the top level. `U-constructor17` is an operator that evokes a rapid reanalysis, but it is no different in kind than any other u-constructor that populates the top problem space.

3.4 The situation model

The situation model serves as both a semantic and referential representation (§2.1). In this section we consider the structure and computation of the situation model, focusing on its role as a semantic representation. The processes of referent resolution are described in §3.5.

3.4.1 What the situation model represents

Many semantic theories in artificial intelligence and linguistics make some commitment to a distinguished level of primitives into which all meaning is decomposed. For example, the Katz & Fodor (1963) theory decomposes meaning into primitive *semantic markers*, and conceptual dependency (Schank & Riesbeck, 1981) analyses events into a fairly small set of abstract schemas.

An alternative approach, and the one taken in NL-Soar, is to assume that the ontology is at least as rich as natural language itself. This leads naturally to a lexically-based semantics. The kind of representation that emerges from such an ontology is in a sense superficial: it closely reflects the particular lexical content of the surface form[9].

The 50,000+ synsets of WordNet (Miller, 1990) provide one possible basis for a lexically-centered semantic ontology. Each synset corresponds to a set of synonymous word senses, and therefore provides a unique category label. Although NL-Soar does not systematically exploit WordNet (or any other ontological resource), WordNet exemplifies the kind ontology assumed in NL-Soar[10].

We further assume as part of the ontology a decomposition of situations into objects, relations, and properties. The semantic domains for each of the three basic elements are unrestricted—anything drawn from the rich ontology is possible.

3.4.2 How the situation model represents

The situation model is an annotated model, in keeping with the basic models assumption in NL-Soar (69). Thus, the situation model consists of objects, properties, and relations that map directly onto the objects, properties, and relations in the represented situation. Figure 3.18 shows how the model is realized in attribute-value structures in working memory. This example illustrates a few basic points about NL-Soar's situation models:

- There is no obligatory primitive decomposition. For example, the property of being a *bachelor*, which could be decomposed into something like *unmarried male*, is retained as a first-class ontological category. This is just a reflection of the lexically-based semantics discussed above. There is no limit to how specific the semantic labels may be—they will be as specific as the utterance requires.

- Although there is no obligatory decomposition, more general properties may be explicitly represented (e.g., the *animate* properties of *bachelor* and *ball*). The lexically-based ontology does not preclude decomposing the specific categories into more general features, if processing demands it.

[9] There is some psychological basis for superficial semantic representations (Simon & Hayes, 1979; Fodor et al., 1975).

[10] The adoption of a lexically-based ontology does not rule out the incorporation of semantic categories with no lexical realization. For example, many of the more abstract classes assumed to comprise the upper portion of semantic hierarchies are missing in WordNet, since these classes are not necessarily labeled by individual words.

```
                (obj2 ^property p1 p2 ^relation r1 r2)
                (p1 ^name isa ^value see2-event)
                (p2 ^name time ^value past)
                (r1 ^name agent ^to obj1)
                (r2 ^name object ^to obj3)

(obj1 ^property p3 p4)                   (obj3 ^property p5 p6)
(p3 ^name isa ^value bachelor1           (p5 ^name isa ^value ball3)
(p4 ^name animate ^value t)              (p6 ^name animate ^value nil)
```

FIGURE 3.18: A possible situation model for *The bachelor saw the ball.* `see2-event` corresponds to one of the meanings of *see* as a verb; `ball3` corresponds to one of the meanings of *ball* as a noun, and so on.

- The situation model as depicted in Figure 3.18 does not encode the *identities* of the objects. It just encodes semantic descriptions which may refer to existing entities. §3.5 discusses the process of referent resolution.

Adopting a mental model-based representation raises many difficult representional issues, such as how to handle quantification, disjunction, abstract concepts, propositional attitudes, and so forth. Although these are difficult problems, they are not insurmountable; Johnson-Laird (1983) sketches solutions to some of these problems. Indeed, the fact that models are not well suited to handling arbitrary disjunction and quantification is a source of explanatory power for the theory in accounting for performance on verbal reasoning tasks. Nevertheless, as with any theory of semantic representation, much work remains to be done. At this stage in its development, NL-Soar simply adopts the models framework without advancing it further as a representational theory.

3.4.3 Constructing the situation model and chunking s-constructors

The situation model is assembled in the S-construct space, the counterpart of U-construct for u-constructors. There are three operators in S-construct, corresponding to the three basic entities in models: *create-referent* establishes a new object, *add-property* augments an existing object with a property, and *add-relation* establishes a relation between two objects. These operators map directly from the utterance model and lexical features to the situation model. The situation model encodes the reference-independent meaning of the utterance; it serves as input to the reference resolution operators.

Figure 3.19 traces the construction of part of the situation model while comprehending *The Magic defeated the Knicks.* We begin just after *defeated* has been projected to a VP and IP, and [$_{NP}$ *the Magic*] has been attached in subject position. First, a create-referent operator establishes a referent for the VP, which will become the object representing the event described by the sentence. Create-referent also generates a referring relation between part of the utterance model (in this case, the VP) and the new object in the situation model. Next, add-property, triggered by the lexical content of the VP, augments the object with the property *defeat4* (labeling some kind of defeat event). Finally, add-relation establishes the *agent* relation between the referent of the subject of the sentence [$_{NP}$ *the Magic*] and the referent of the VP. This completes the situation model for the fragment [$_{IP}$ [$_{NP}$ *the Magic*] [$_{VP}$ *defeated*]].

New s-constructors emerge in a manner similar to u-constructor creation. The problem space hierarchy for building s-constructors is shown in Figure 3.20. The hierarchy parallels the structure for building u-constructors (with one important difference which we will discuss in §3.6). Like u-constructors, s-constructors are assigned gensym constant names. Chunking over the deliberate construction of the situation model produces proposal and implementation associations for new s-constructors. For example, the problem solving in Figure 3.19 produces a proposal association for a new s-constructor (`s-constructor23`):

(74)
IF there is a VP headed by *defeat*
and its IP projection has an NP in specifier position
and the NP has a referent
THEN propose `s-constructor23`

The example in Figure 3.19 illustrates just the simplest kind of situation model construction. In general, building the situation model may require drawing on task-specific knowledge sources and engaging in considerable problem solving. Although such processes are not an implemented part of NL-Soar, the basic architecture supports them. The additional knowledge sources may be added in the S-construct space, or in spaces below S-construct. Whatever form the knowledge takes in the lower spaces, the architecture (particularly chunking) ensures that the system continually makes the shift from deliberation to recognition, so that over time even some fairly task-specific interpretation knowledge may be brought to bear as part of recognitional comprehension. This process is called the *taskification* of language (Lehman, Newell, Polk & Lewis, 1993b).

What semantic features should lexical access retrieve?

Chunk (74) clearly raises an important issue for semantic interpretation in NL-Soar: what set of semantic features should be retrieved by lexical access? Proposal association (74) actually tests for the specific lexical item (the uninflected root). Thus, the present organization represents the extreme hypothesis that lexical access per se initially retrieves *no* semantic features. In effect, the semantic access of the lexicon is accomplished first via the add-property and add-relation operators, and after chunking via the s-constructors. Although this extreme organization is probably wrong, the alternative organization that simply selects from a precomputed set of semantic senses is also probably wrong (e.g., (Pustejovsky & Boguraev, 1993)).

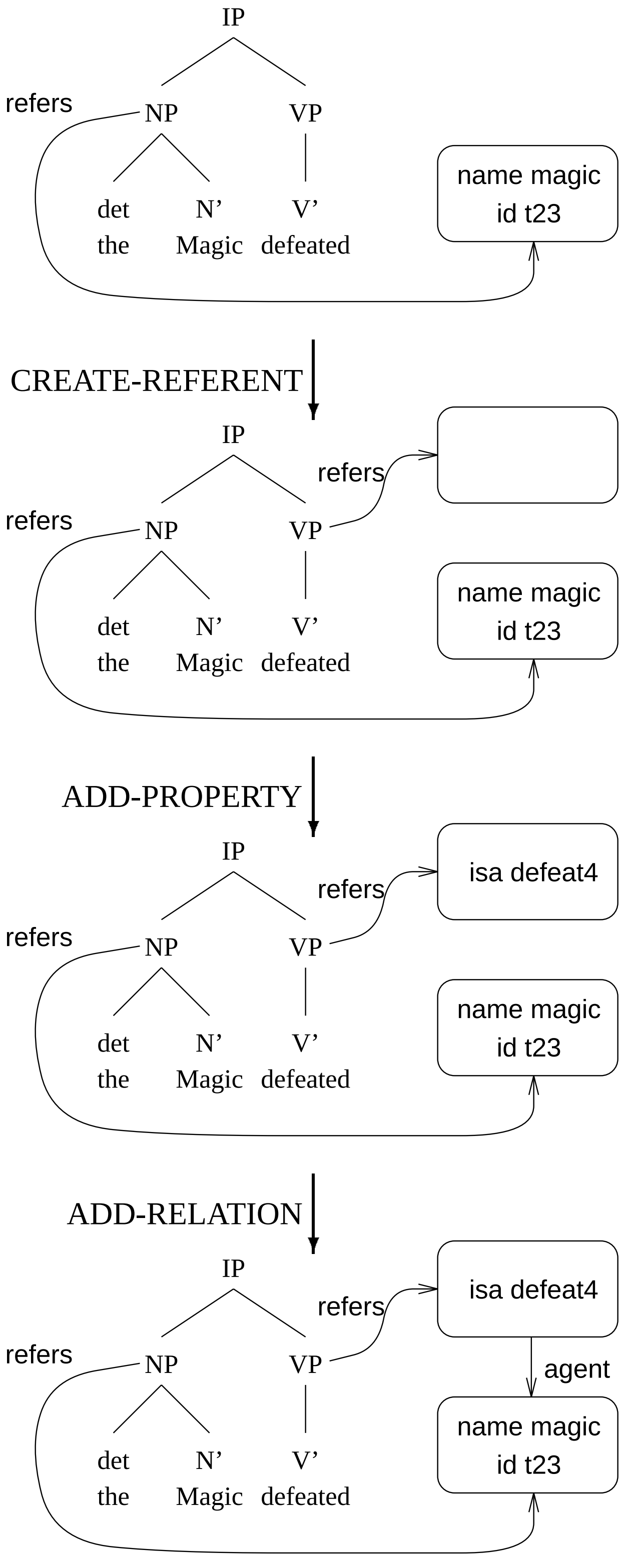


FIGURE 3.19: Constructing the situation model. The *create-referent* operator creates a new object in the situation model and establishes the reference link from the utterance model. The new object is given content with *add-property* and *add-relation* operators.

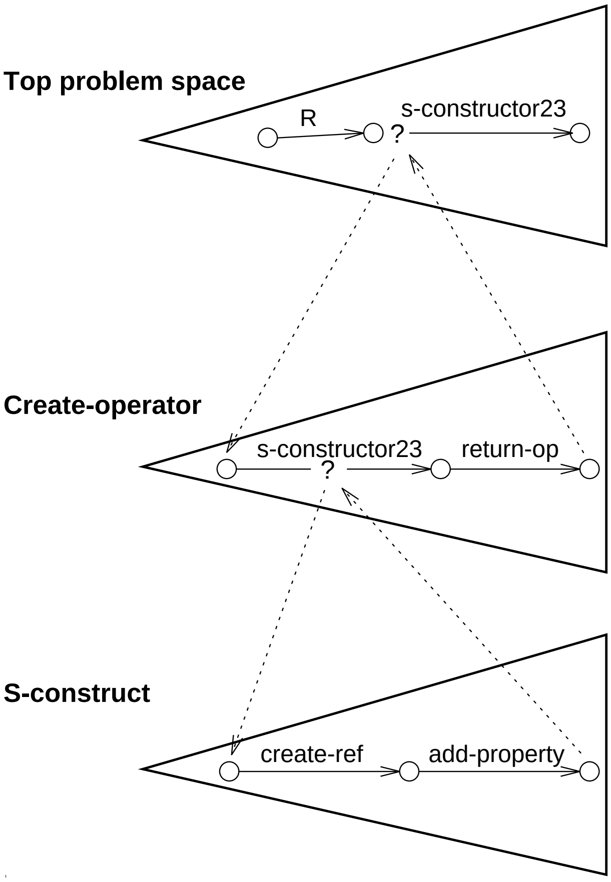


FIGURE 3.20: The problem spaces for building s-constructors. *Create-operator* generates new operator names, and *S-construct* contains the primitive situation model construction operators.

Repairing the situation model

Situation model repair follows on the heels of utterance model repair. Unlike the U-construct space, however, there is no additional destructive operator in S-construct. As the utterance model evolves, it triggers new s-constructors that keep the situation model current. When the new structure conflicts with part of the existing structure, the older structure is simply destroyed as part of the process of establishing the new piece of the model.

This minimalist approach to repairing the situation model is not guaranteed to remove all extraneous or out-of-date structure from the situation model. For example, in the repair of *The square table* (Figure 3.15), the evolution of the utterance model from [$_{NP}$ *the square*] to [$_{NP}$ *the* [$_{AP}$ *square*] *table*] will give rise to two situation model objects, namely, a square and a table. The square will not be destroyed upon creating the square table, since at the level of the situation model there is nothing about the table that is incompatible with the existence of the square. However, the square will remain disconnected from the extended situation model, since the remainder of the utterance is about the table. Thus, if the sentence continued *the square table is too large for my living room*, NL-Soar would not be confused into thinking that *the square* is too large.

Although this repair scheme looks promising, it is still nascent. Further experience with extended texts will be required to establish its viability.

General features of s-constructors

The examples of situation model construction described above illustrate several general features of s-constructors in NL-Soar, many shared with u-constructors:

- A single s-constructor may be composed of many primitive structure building operations (e.g., add-property).
- There is no fixed vocabulary of s-constructor types. The variety of s-constructors is a function of the indefinite variety of structures potentially produced by the S-construct space.
- The content of the operator is not specified by explicit data structures which must be interpreted. In particular, there is no intermediate mapping language between the utterance model and situation model. Semantic interpretation knowledge is essentially held in the proposals for situation model constructors (both the s-constructors in the top space and the primitive operators in the S-construct space).
- The repair of the situation model is directly triggered by the *results* of the repair process in the utterance model. The situation repair is an integrated part of the s-constructor.
- The construction of the situation model is an incremental process that works on partial syntactic structure. S-constructors do not wait for the completion of any particular syntactic node before they begin interpretation.

3.5 Reference resolution

Reference resolution, like semantic interpretation, can require arbitrary amounts of inferencing. However, we know from the evidence in §2.2.3 that much reference resolution is a rapid process. Furthermore, reference resolution must make contact with long-term memory as efficiently as short-term memory. The referents of noun phrases such as *CMU* are quickly ascertained regardless of whether the object has been recently established in the discourse[11]. NL-Soar's reference resolution mechanism is primarily concerned with explaining this rapid resolution to objects in both short- and long-term memory.

[11] How one resolves *CMU*, of course, depends on how much time one has spent in central Michigan.

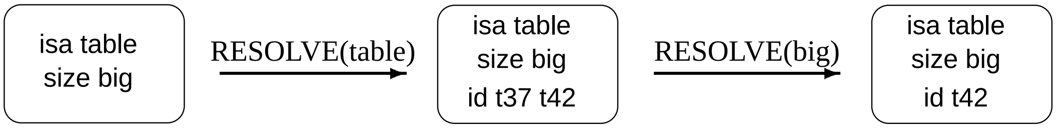


FIGURE 3.21: Resolve operators recognize aspects of the situation model as semantic descriptions of known objects. The recognition produces a constant symbol that is the identifier of the object in long-term memory.

3.5.1 Reference resolution as recognition

NL-Soar takes reference resolution to fundamentally be a process of *recognition*. Reference resolution recognizes pieces of the situation model as partial descriptions of known objects. The process augments the situation model with constant identifiers which are symbols denoting the object in long term memory (that is, knowledge about the object is held in associations in long term memory that have the symbol as part of their conditions or actions).

Thus, there is no representational level between the syntax and the referential representation: the situation model *is* the referential representation, or more accurately, the *resolved situation model* is the referential representation. This organization contrasts with multi-level models such as READER (Thibadeau et al., 1982), which posit independent levels of semantic and referential encoding. In effect, the situation model starts as a semantic (sense) encoding and evolves into a referential encoding.

Figure 3.21 shows a simple example, resolving the noun phrase *the big table*. Suppose there are two tables in the discourse, only one of which is big. The big table has the LTM identifier `t42`, and the other table is `t37`. First, a resolve operator is applied to the new situation model object, instantiated for the property `isa table`. This operator triggers two associations in LTM that retrieve `t37` and `t42` as identifiers of objects that satisfy this part of the description. The situation model object is augmented with both identifiers.

Next, a resolve operator is applied to the object with respect to the property `size big`. This operator triggers an association that recognizes `size big` as a property of `t42`. Since `t37` is not so recognized, it is removed as an augmentation of the object. The result is the unique identification of *the square table* as `t42`. Had this part of the description been insufficient to uniquely identify the referent, both identifiers would have persisted, awaiting for incoming material to pare down the set (e.g, a post-nominal PP like *in the bedroom*). This kind of incremental approach is similar to the one used in SHRDLU (Winograd, 1972) (see also (Altmann & Steedman, 1988)).

3.5.2 Building recognition chunks

Where do these reference resolution associations come from? Figure 3.22 illustrates what happens when the resolve operator is applied to the `isa table` property, and associations fail to retrieve any object identifiers. An implementation impasse arises, and NL-Soar takes this as a signal that the present description introduces a new object into the discourse. The impasse provides the opportunity to learn new recognition chunks for the description. Soar

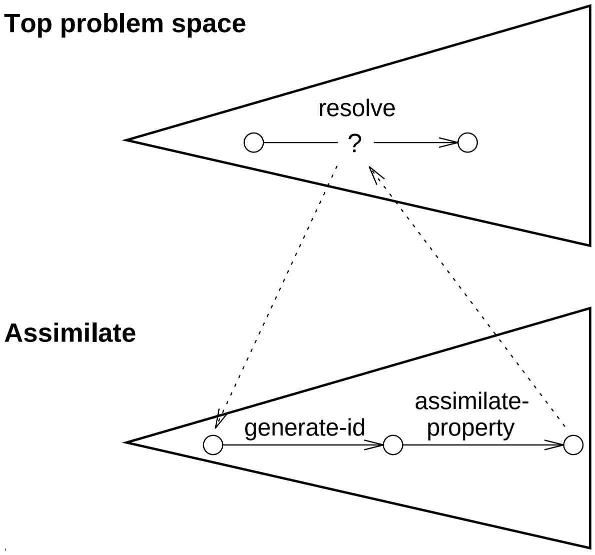


FIGURE 3.22: Learning new reference resolution recognition chunks. When a new piece of the situation model is not recognized, an impasse arises on a resolve operator. In the *Assimilate* space, the system learns to recognize the novel structure by associating its components (properties and/or relations) with a newly generated constant identifier.

drops into the *Assimilate* space and creates a new constant identifier to associate with the new object. Next, an *assimilate* operator is applied to the attribute and value pair associated with this property. Based on the recognition of the property, the new identifier is returned to the higher space. The result is a chunk of the form:

(75)
IF applying the resolve operator to a situation object
with respect to the property `isa table`
THEN the identifier of the object is `t37`

Similar processing produces the chunks that associate properties with established objects. For example, associating *big* with the object `t37` produces the chunk

(76)
IF applying the resolve operator to a situation object
with respect to the property `size big`
and the object has identifier `t37`
THEN `t37` is recognized as having this property

Such chunks can be used to narrow down the referent set as described above.

This use of chunking provides a clear example of chunking as knowledge-level acquisition; speedup learning is not at issue here. The system is not simply learning to recognize something faster. The recognition chunk's existence encodes the fact that the system has seen the object before (Rosenbloom, Newell & Laird, 1991).

Recognition chunk (75) is overgeneral. If chunks like this were always produced, identifiers for all the tables ever encountered by the system would be retrieved when resolving *table.* Additional contextual discrimination is required to restrict the pool of potential referents to a manageable set. The discourse segment stack of Grosz & Sidner (1986) is one kind of mechanism that uses the structure of the discourse itself to provide the necessary discrimination. In NL-Soar, this is partially realized by having the recognition process associate the new identifier not only with a particular property or relation, but also with some representation of the current discourse segment. If the discourse segment is represented by a constant identifier on the state, then chunk (75) would look something like:

(77) *IF* applying the resolve operator to a situation object
with respect to the property `isa table`
and the discourse segment is `s98`
THEN the identifier of the object is `t37`

This is just barely a beginning of a theory of discourse structure in NL-Soar, and it raises additional questions, such as: How are the discourse segment symbols retrieved? How are transitions made from segment to segment? While a theory such as Grosz and Sidner's provides some help, the implementation within the constraints of Soar is nontrivial. For example, one cannot simply directly incorporate the stack structure. Soar's recognition memory is not a stack.

3.5.3 Long term memory of content: reconstructive

The process of reference resolution in NL-Soar has an important side effect: the recognition chunks constitute a long term memory of the content of the discourse. The memory is recognition-based, not recall-based. NL-Soar does not perfectly memorize the content of each discourse. It cannot recall arbitrary parts of the discourse based on arbitrary cues. Rather, memory retrieval is necessarily a *reconstructive* process in which other knowledge sources must be used to compose structures that trigger the recognition chunks.

A simple example will illustrate. Suppose Soar has comprehended the utterance *The dog is a Spaniel.* Reference resolution will build two chunks of the form

(78) (a) `isa dog` → `o23`
(b) `o23, breed spaniel` → recognized

Suppose at a later time Soar must answer the question *What breed is the dog?* Assuming that referent resolution correctly identifies the *the dog* as `o23`, Soar must retrieve the breed associated with `o23`.

The problem is that chunk (78b) is not a recall chunk. In other words, it does not map `o23` to `spaniel`. Soar must engage in a generate and test in an attempt to trigger the chunk. Figure 3.23 shows one possible realization of the process. An impasse arises (1) when the answer to the question is not immediately retrieved. Soar enters a space to reconstruct the answer, via a series of resolve operators that generate plausible candidates for the dog's breed (based on general knowledge about dogs). Each resolve operator is of the same form

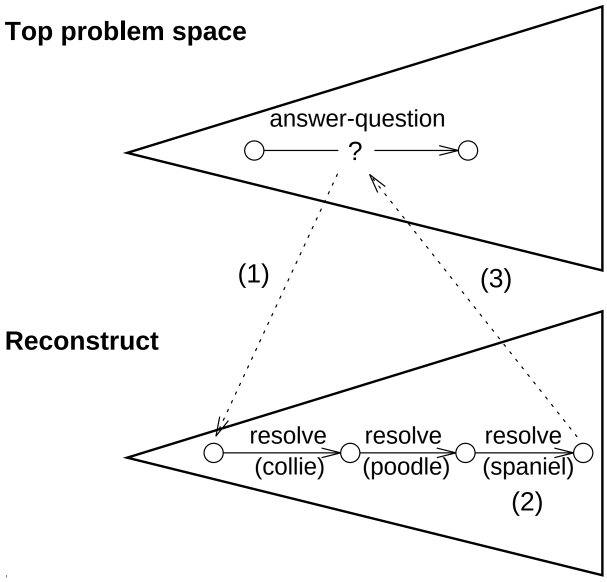


FIGURE 3.23: Recall by reconstruction. The system is attempting to remember what breed a particular dog is. The method is generate-and-test. Plausible candidates are generated via general knowledge sources, and the correct answer triggers the recognition chunk.

as the resolve operators used during comprehension: each associates a particular property (in this case, a dog breed) with a particular object (in this case, a situation model object identified as `o23`). When the correct choice (spaniel) is generated, chunk (78b) fires (2) and the goal is achieved. Spaniel is returned as a result of the problem solving (3), building a chunk of the form

(79) `o23`, breed? → `spaniel`

As a result of the reconstructive problem solving, Soar now has the knowledge available in a new form: a recall chunk.

This generate-and-test behavior is part of the solution to the data chunking problem in Soar, which is essentially the problem of using chunking to store away declarative knowledge in a usable form in the recognition memory (Newell, 1990; Rosenbloom et al., 1991). The success of any reconstructive process depends on finding good generators to limit the search (e.g., the space of dog breeds used above). The general problem may seem intractable, but a number of successful Soar systems have been implemented which use some form of recognition-based memory and generate-and-test-based retrieval (Lehman & Conati, 1993; Vera, Lewis & Lerch, 1993; Huffman, 1993). In these models, the key has been to use the external environment or current problem solving context in combination with task problem spaces to constrain the generation process. The simple system described in (Vera et al., 1993) takes instructions for using an automated teller machine and reconstructs the instructions based on the machine's affordances. The system of Huffman (1993), which takes instructions for a robot arm manipulating a blocks world, actually uses NL-Soar for the natural language component and reconstructs the instructions from precisely the kind of reference resolution chunks described above. The reconstruction is guided by the existing task knowledge in the system.

In short, what NL-Soar can recall about a discourse is a function of what else it already knows about the objects of the discourse, as well as the cues that might be available in the current situation. Our practice in building Soar systems with reconstructive memories indicates that when the reconstruction process is a goal-driven, situated activity, natural generators can often be found to keep the search from becoming prohibitively expensive.

3.5.4 General features of reference resolution

The essential features of NL-Soar's reference resolution can be summarized as follows:

- Reference resolution is a process of recognizing parts of the situation model as descriptions of previously encountered objects. The process resolves situation model objects to constant identifiers in long term memory that denote known objects. The process does not resolve syntactic phrases, and it does not resolve to other situation model objects or previous syntactic phrases.

- The process is incremental. It works on partial descriptions and narrows the pool of referents as information becomes available.

- The process builds associations which form a recognition memory of the content of the discourse.

- The associations are fine-grained in nature, so that the information associated with any particular object is distributed in long term memory across many chunks (Miller, 1993).

- Only what is novel in a discourse is stored, because opportunities to learn new chunks arise only when recognition of the content fails.

- Later retrieval of the content by other problem solving is a reconstructive process in which existing knowledge sources and the current situation and problem solving context play an important role.

The currently implemented processes are fairly simple, and much work lies ahead to incorporate richer forms of inferencing to determine reference, discourse segment structures, and so on. However, as the list above makes clear, even this basic structure has some interesting properties, and its functionality has been demonstrated in at least one application (Huffman, 1993).

3.6 The control structure of comprehension

What is the control structure of comprehension? This question is at the heart of the psycholinguistic debate on modularity (though it is rarely cast in those terms). However, the field has been mostly concerned with what *content* guides the flow of processing, with less attention to making explicit mechanistic proposals for control structure.

NL-Soar provides a well-defined answer: The control structure of comprehension is the open, recognize-decide-act control structure of the Soar architecture (§3.1.2). This has a number of immediate implications:

- Each decision in language processing is *potentially* open to any knowledge source—syntactic, semantic, or contextual. There is no restriction on what aspects of the current problem space context may be tested by the associations that comprise the comprehension operators.

- The decisions are not fixed in advance, but the knowledge is assembled at the moment of the decision (via the elaboration phase of the decision cycle). The control flow is a function of whatever knowledge is *immediately* available (via the recognition match) at the time the decision is made.

- The control knowledge is open to continuous modification via chunking. Any decision point in NL-Soar can potentially be modified if new associations are learned and brought to bear at the appropriate time.

In the next few sections, we will explore the consequences of Soar's control structure for ambiguity resolution and functional parallelism.

3.6.1 Syntactic ambiguity

Syntactic ambiguity manifests itself when multiple u-constructors are proposed in parallel at a single point during comprehension. For example, consider the local ambiguity present in (80):

(80) The doctor told the patient *that* . . .

The CP headed by *that* can be attached as the second complement of *told* or as a relative clause modifying *patient.* Suppose that the operator that attaches second sentential complements is `u-constructor7` and the operator that forms NPs modified by a relative clause is `u-constructor2`. Then both operators will be proposed upon encountering *that* in (80) (Figure 3.24).

How should this ambiguity be resolved? The Altmann, Crain, and Steedman work (§2.3.5) provides a knowledge-level theory for resolving ambiguities of this type: the attachment site should depend on the success or failure of the simple NP (in this case, [$_{NP}$ *the patient*]) to uniquely refer in the present context. If the simple NP fails to select a

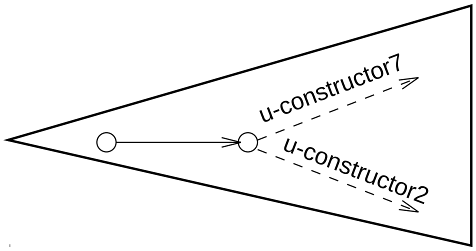


FIGURE 3.24: Syntactic ambiguity manifested as multiple u-constructors. At a local structural ambiguity, multiple u-constructors (two or more) may be proposed in parallel. Each u-constructor corresponds to a unique structural interpretation.

single referent, then choose the restrictive relative clause attachment, since that may provide additional information to narrow the set of referents.

The information required to implement this strategy is present in the utterance and resolved situation models because reference resolution is an incremental process in NL-Soar, as described in §3.5.1. A search control association can test the proposed operators, the current utterance model, and the referential status of the situation model as follows:

```
           IF   operators u-constructor7 and u-constructor2 are proposed
                and the NP in the assigners set refers to a situation model object
(81)            and the situation object currently has more than one identifier
                  associated with it
           THEN  prefer u-constructor2 to u-constructor7
```

Figure 3.25 shows what happens if this chunk is in recognition memory and the context is such that the simple NP refers to more than one individual. Within a single decision cycle, both u-constructors are proposed in parallel, triggering chunk (81). The result of the decision cycle is to select `u-constructor2`, corresponding to the relative clause reading.

Similar control associations can be based on semantic content, rather than referential context. In Chapter 4 we will see how such associations can arise from chunking, and discuss the implications of this model with respect to modularity.

3.6.2 Semantic ambiguity

Semantic ambiguity manifests itself when multiple s-constructors are proposed in parallel at a single point during comprehension. For example, consider the lexical ambiguity present in (82):

(82) That pudding is *rich*.

Rich is interpreted differently depending on what thing it is modifying. There are two senses present in the system: one sense labels something as rich to eat, and the other sense labels something as financially wealthy. Suppose that the s-constructors which

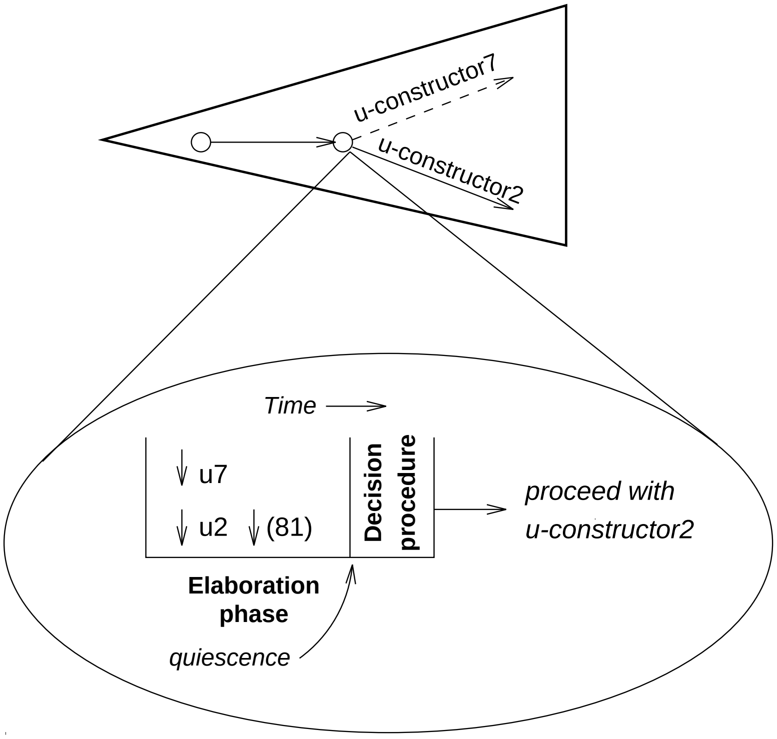


FIGURE 3.25: Resolving syntactic ambiguity recognitionally. The structural alternatives (`u7` and `u2` are proposed in parallel. This triggers an association (chunk 81) that encodes knowledge about which path is preferred.

interpret the predicate complement construction with *rich* are `s-constructor97` (food) and `s-constructor44` (wealthy). (The set of s-constructors evoked by a particular word do not necessarily map one-to-one to a set of senses of that word.) As in the case of syntactic ambiguity, both operators are proposed in parallel at *rich.* The semantic preference for one sense over the other is captured by the association:

(83)
IF `s-constructor97` and `s-constructor44` are proposed
and referent of the subject has the property `isa food`
THEN prefer `s-constructor97`

There is another possibility for resolving semantic ambiguity: build semantic constraints (selectional restrictions) into the generator for the s-constructors. In such a scheme, `s-constructor44` would not be proposed in (82) because the requirement that the wealthy sense of rich modify a person is not met. This could be implemented via a series of constraint checks in a manner similar to the syntactic constraint checking. In fact, earlier versions of NL-Soar did take such an approach (Lehman et al., 1991a).

The problem with this approach is that semantic constraints are inherently preferential—they are not absolute filters. This has been pointed out by Wilks (1975) and others. The generation of multiple alternatives allows knowledge encoded as preferences to be brought to bear. Furthermore, no matter how much of the knowledge is encoded into the generator,

TABLE 3.2: Mapping comprehension functions onto problem space functions.

COMPREHENSION FUNCTION		PROBLEM SPACE FUNCTION
Parsing	Lexical access	Encoding
	Generation of syntactic alternatives	Operator proposal
	Selection among syntactic alternatives	Search control
	Construction of syntactic structure	Operator application
Interpretation	Generation of semantic alternatives	Operator proposal
	Selection among semantic alternatives	Search control
	Construction of semantic representation	Operator application
Reference resolution		Operator application

there is always the potential for an ambiguity to arise that is not filtered out by the fixed generators. In that case, there is no recourse but to resolve the ambiguity via control associations similar to (83). The general issue of how knowledge should be distributed across problem space functions (proposal, selection, implementation of operators) will be addressed systematically in §3.7.

3.6.3 Functional parallelism

Parallelism is an inherent part of NL-Soar because Soar's recognition memory is a parallel system. This parallelism arises in two ways. The match process itself is massively parallel because all associations are continuously matched. And, once matched, the associations fire in parallel.

A more informative characterization of parallelism in NL-Soar is made possible by breaking down the comprehension process into functional components and analysing the parallelism in functional terms. Table 3.2 shows the functions of comprehension and their mapping onto problem space functions (encoding, operator proposal, operator selection, and operator application) in NL-Soar. Using this mapping, we can consider parallelism *within* function, and parallelism *across* functions.

Parallelism within functions

Parallelism within function refers to the simultaneity of distinct processes that realize a particular comprehension function. In the top level space, each of the functions can be realized within a single decision cycle. The decision cycle is the level of serial control in Soar. Within the decision cycle, the recognition memory fires associations in parallel and in sequence. The parallelism is limited only by the inherent data dependencies in the processing. Thus, there is the potential for parallelism in all of the comprehension functions. In fact, there *is* parallelism within all of the functions, described in the following list:

- In *lexical access*, the associations that represent the different categorial senses of a word fire in parallel.

- In *syntactic and semantic generation*, the associations that propose the applicable comprehension operators (u-constructors and s-constructors) fire in parallel.

- In *syntactic and semantic selection*, the associations that evaluate the alternative operators fire in parallel. For example, suppose that three operators are proposed U_1, U_2, and U_3. The associations establishing the preferences $U_1 > U_3$ and $U_2 > U_1$ may fire in parallel.

- In *syntactic construction*, the primitive structure building associations which compose u-constructors may fire in parallel. For example, in forming the noun phrase [$_{NP}$ *the red block*], the associations which adjoin the adjective phrase [$_{AP}$ *red*] to [$_{N'}$ *block*] can fire in parallel with the associations which project [$_{N'}$ *block*] to the maximal projection NP.

- In *semantic construction*, the primitive structure building associations which compose s-constructors may fire in parallel. For example, in forming the situation object corresponding to the newly created NP [$_{NP}$ *the red block*], the associations establishing the properties `color red` and `isa block` fire in parallel.

- In *reference resolution*, the associations that recognize parts of the situation model retrieve identifiers of potential referents in parallel.

Parallelism across functions

Parallelism across functions refers to the simultaneity of processes that realize different comprehension functions. The one-at-a-time application of operators in Soar imposes architectural limitations on functional parallelism. However, the serial stream of operators only restricts parallelism in operator application—proposal and selection may go on in parallel with each other and with application.

In NL-Soar, this means syntactic structure building, semantic structure building, and reference resolution cannot happen in parallel. But the functions of lexical access, syntactic and semantic generation, and syntactic and semantic selection all may happen in parallel. Figure 3.26 shows this graphically. The horizontal axis represents time. The top three boxes represent the seriality of structure building and reference resolution. Any vertical slice through the figure represents a set of functions that can potentially happen in parallel during the same decision cycle.

As we can see from Figure 3.26 and the analysis of within-function parallelism, NL-Soar's control structure yields a mix of serial and parallel processing. Parallelism within function is limited only by inherent data dependencies. Parallelism across function is limited by the seriality of Soar's decision cycle, which restricts simultaneity of operator application. Yet at any given moment, certain aspects of syntactic, semantic, and referential processing may be happening in parallel.

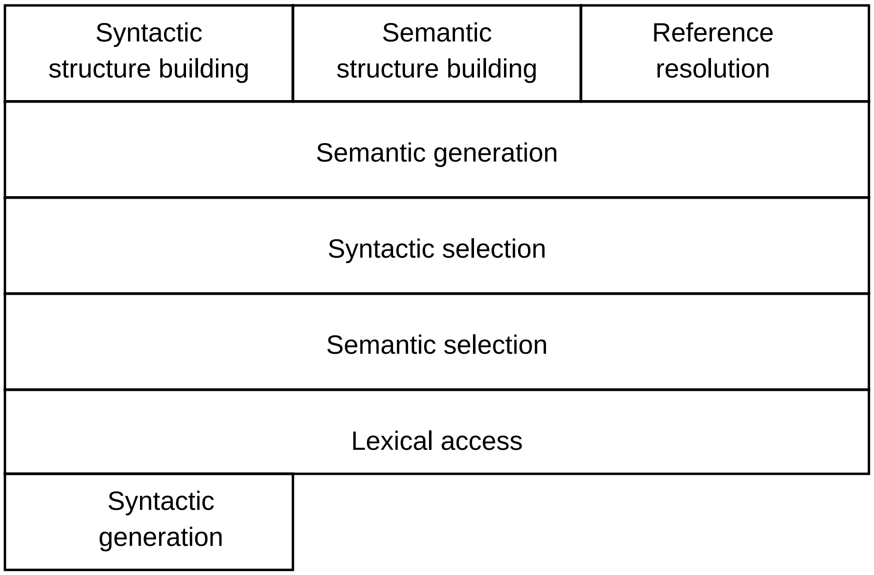


FIGURE 3.26: Parallelism across functions in NL-Soar. Time is along the horizontal axis. Any vertical slice through the figure cuts through functions that may happen in parallel.

3.7 Evaluating a space of alternative models

The previous sections assumed a particular structure for comprehension operators: the operators come in three types (u-constructors, s-constructors, and resolve operators), and the constructors have an unbounded set of tokens (`u-constructor7`, `u-constructor44`, etc.) Now that the nature of this structure has been explored fairly thoroughly, we can step back and examine the space of alternative comprehension operator schemes and the motivation for the present model.

3.7.1 The space

The following five parameters define a space of NL-Soar models by specifying various aspects of comprehension operators. The parameter values simply enumerate the architecturally permissible ways of realizing comprehension operators. The parameter values are not all independent, an issue to be explicitly addressed later in this section.

P1 Comprehension operator types

This classification is by output—what models are produced. U = utterance model, S = situation model, R = resolved situation model. For a given word, any subset of operator types may be evoked to perform comprehension. (The model discussed in this chapter corresponds to value (d)).

(a) → U S R

(b) → U S, → R

(c) → U, → S R

(d) → U, → S, → R

P2 Operator tokens per type

If there is one comprehension token per type, then that token is always selected and applied to each word. If there are multiple tokens per type, then each token may be evoked by various aspects of the current comprehension context. (The present model corresponds to value (b)).

(a) One operator token per type

(b) Many operator tokens per type

P3 Detection and representation of ambiguity

Since true local ambiguity is unavoidable, there must be some way to detect and momentarily represent the ambiguity so that knowledge may be brought to bear on the choice. The possibilities for detecting and representing ambiguity include two architectural mechanisms. (The present model corresponds to value (a)).

(a) Context slots (e.g., operators, states) [architectural]

(b) Attribute impasses [architectural][12]

(c) Special data structures [non-architectural]

P4 Distribution of knowledge across proposal, selection, and implementation of comprehension operators

The multiple knowledge sources that must be brought to bear in comprehension may be distributed in different ways across the functions of proposing, selecting, and implementing operators. (The present model distributes most of the knowledge in proposing and selecting operators; for example, all syntactic constraints are chunked into the operator proposals. This corresponds to value (c).)

(a) Most knowledge in implementation

(b) Most knowledge in proposal

(c) Most knowledge in proposal and selection

(d) Distributed across all three

P5 Initial lexical access

The initial retrieval of lexically specific knowledge may be context dependent (moving knowledge to the generator) or context independent (making the generator knowledge-lean). (In §3.8 it was noted that experimental evidence seems to favor context independent access. The present model takes this route, corresponding to value (b)).

[12] See the Soar manual (Laird et al., 1993) for an explanation of attribute impasses.

TABLE 3.3: Dependencies in the parameter space.

P2a→P3b/c and P4a
P2b→P3a and P4c/d
P3a→P2b and P4b/c
P3b→P4a/d
P3c→P4a/d
P4a→P3b/c
P4b→P2b and P3a
P4c→P2b and P3a
P4d→P2b and P3a

(a) Context dependent

(b) Context independent

As mentioned above, the parameters are not completely independent—selecting certain values for certain parameters fixes the settings for other parameters. In particular, P2, P3 and P4 are all interdependent. For example, having many tokens per operator type forces knowledge to be in the proposal and selection of comprehension operators (P2b → P4c/d). The complete set of dependencies is given in Table 3.3.

3.7.2 The evaluation

The models in this space will be evaluated against three basic computational and functional criteria:

C1 *Transfer.* Evaluates schemes based on how they affect the specificity of the chunks that realize recognitional comprehension in the top space. Along this dimension, schemes that build more general chunks are rated better than those that build specific chunks.

C2 *Asymptotic efficiency.* Evaluates schemes based on the maximum achievable efficiency of recognitional comprehension. That is, comprehension *after chunking*, assuming chunks transfer. The less recognitional operators, the more efficient the scheme.

C3 *Character of implementation.* Evaluates schemes based on the simplicity and economy of the data structures and processes. This is of course a subjective measure, but it is often possible to make clear qualitative distinctions. This criterion essentially evaluates alternatives based on how naturally they fit in the Soar architecture. The more layers of mechanism required to implement a solution, the less natural the solution is.

TABLE 3.4: Independent evaluation of parameter/value pairs.

	C1: Transfer	C2: Asymptotic efficiency	C3: Simplicity
P1a		+	○
P1b		○	+
P1c		○	○
P1d		–	+
P2a			
P2b			
P3a			+
P3b			–
P3c			–
P4a			
P4b	–	+	
P4c	+	–	
P4d			
P5a	–	+	
P5b	+	–	

Relative evaluation: + better than ○ better than –
No entry means no effect

The total efficiency of the system is a function of the transfer rate (C1), recognitional efficiency (C2), and the efficiency of deliberate comprehension. Deliberate comprehension efficiency is not an evaluation criterion, since the schemes do not differ along this dimension.

Table 3.4 gives an independent evaluation of each parameter/value pair with respect to each criterion. This is a simple direct evaluation, not taking into account the dependencies noted in Table 3.3. From Table 3.4 and the dependencies in Table 3.3, we can compute a complete independent evaluation, shown in Table 3.5.

From this evaluation, P1a, P1b and P1d is preferred to P1c; P2b is preferred to P2a; P3a is preferred to P3b and P3c; P4b and P4c is preferred to P4a and P4d. We can reevaluate the parameters in the restricted space, fixing the parameters P1={a,b,d}, P2=b, P3=a, P4={b,c}, and P5={a,b}. Table 3.6 gives the results. The preferred system corresponds to: P1 = d, P2 = b, P3 = a, P4 = c, and P5 = b which is the model presented in this chapter. Thus, the basic structure of comprehension operators and lexical access is not an arbitrary choice, but one guided by the functional implications of implementing various sets of mechanisms within the Soar architecture.

Table 3.5: Evaluation of parameter/value pairs, taking into account the dependencies.

	C1: Transfer	C2: Asymptotic efficiency	C3: Simplicity
P1a		+	○
P1b		○	+
P1c		○	○
P1d		–	+
P2a			–
P2b			+
P3a			+
P3b			–
P3c			–
P4a			–
P4b	–	+	+
P4c	+	–	+
P4d			+
P5a	–	+	
P5b	+	–	

Relative evaluation: + better than ○ better than –
No entry means no effect

Table 3.6: Evaluation of parameter/value pairs in restricted subspace.

	C1: Transfer	C2: Asymptotic efficiency	C3: Simplicity
P1a	–	+	–
P1b	○	○	+
P1d	+	–	+
P4b	–	+	
P4c	+	○	
P5a	–	+	
P5b	+	○	

Relative evaluation: + better than ○ better than –
No entry means no effect

3.8 Summary of the theory

The following is a summary of the basic principles of NL-Soar. No attempt is made in this list to clearly separate the contributions of the Soar architecture; that issue is dealt with in Chapter 9.

1. *Comprehension operators.* NL-Soar comprehends language incrementally by applying a series of comprehension operators to the incoming input. There are three kinds of operators: u-constructors, which build a syntactic representation, s-constructors, which build a semantic representation, and resolve operators, which perform reference resolution. The constructors may be composed of multiple primitive structure building operations, and there is no fixed limit on the vocabulary of possible operators. Each constructor is denoted by a unique constant symbol; the processes are not represented by data structures which must be interpreted to yield behavior. Operators take on the order of 50 ms to complete.

2. *Comprehension as a continually improving mix of deliberate and recognitional behavior.* Given the real-time constraints, comprehension must proceed mostly by recognition. When the required knowledge is not immediately available, NL-Soar falls into problem spaces that carry out the comprehension functions deliberately, bringing together independently represented knowledge sources. As a result of this problem solving, NL-Soar automatically learns new associations that directly accomplish comprehension, continually shifting Soar from deliberation to recognition. (The model does not specify the top-level associations, only the lower problem spaces.)

3. *Model representation of syntax, meaning, and reference.* Problem space states in NL-Soar are annotated models (pure models obeying structure correspondence, with annotations of limited scope which increase the representational power or help control processing) representing one particular situation. Comprehension operators build two kinds of model in working memory. The utterance model represents the X-bar phrase structure of the utterance. The situation model represents the particular situation that the utterance is about, decomposed into objects, properties, and relations drawn from a rich ontology.

4. *Limited syntactic index for utterance model.* The nodes in the utterance model are indexed in working memory by their potential syntactic relations, in a structure called the A/R set. Each assigning or receiving relation indexes at most two nodes. All processes, including semantic interpretation, access the utterance model via the A/R set.

5. *Context-independent, parallel lexical access.* Initial lexical access retrieves all categorial senses of a word in parallel, independent of the present syntactic or semantic context. The results of lexical access are bar-level zero nodes which are placed in the A/R set.

6. *Head-driven, constraint-based construction of utterance model.* The construction of the utterance model is a head-driven process which begins with projection of nodes from incoming lexical heads. There are no explicit expectation structures. In the lower problem spaces, independent syntactic constraints are applied to check the well-formedness of putative structural links. There are no explicit phrase structure rules; syntactic structure emerges from the interaction of lexical projections with the independent constraints. The generate-and-test problem solving produces chunks that integrate the multiple constraints.

7. *Simple destructive repair mechanism.* Incoming input that is inconsistent with the current utterance model can result in a momentary parallelism of structure. The inconsistency is repaired by a simple destructive repair mechanism. The mechanism consists of the snip operator, which breaks a structural link in the utterance model, and the existing link operators, which perform the reconstruction. The generation of snip is highly constrained. It is only proposed in two cases: when competing syntactic senses of the same lexical token have been incorporated into the utterance model, and when a structural inconsistency is detected local to some maximal projection.

8. *Reference resolution as recognition of semantic descriptions.* Reference resolution in NL-Soar is a recognition process. Resolve operators are applied to parts of the situation model in an attempt to recognize the model as a semantic description of a known object. The content of the discourse is held in long term recognition memory, which arises automatically from an assimilation process that is evoked when recognition fails. Memory for content is necessarily a reconstructive process which attempts to trigger the recognition chunks. This process is driven by a combination of the immediate situation and existing task knowledge.

9. *Open, mixed parallel/serial control structure.* The control structure of NL-Soar is open. Any knowledge source may be brought to bear to modulate the flow of processing—if the knowledge is immediately available in the recognition memory. The control knowledge is open to continual modification via chunking. The control structure admits a mix of parallel and serial processing. There is parallelism within every comprehension function, limited only by inherent data dependencies. There is parallelism across all comprehension functions, with the exception of the application of comprehension operators, which occurs in a serial stream.

Chapter 4

Structural Ambiguity Resolution

BRITISH LEFT WAFFLES ON FALKLANDS
— Newspaper headline

THIS CHAPTER describes how NL-Soar accounts for some of the major phenomena surrounding structural ambiguity resolution. Garden path effects are not discussed—that is the subject of Chapter 6. Here we focus on the processes of ambiguity resolution per se.

The review of the empirical literature in Chapter 2 revealed that the phenomena of ambiguity resolution are fairly complex. There is evidence for interactive effects across a range of syntactic constructions and context types. There is also evidence for modular effects—the failure to bring to bear certain knowledge sources on-line—across a range of constructions. Of those structural parsing preferences so far proposed, some form of right association and lexical argument preferences appear to be the most robust, in both linguistic analyses of corpora and in behavioral studies.

The next two sections demonstrate how NL-Soar accounts for both modular and interactive effects, drawing directly from the structure of the model presented in Chapter 3. The final section summarizes the NL-Soar theory of ambiguity resolution, and draws some general conclusions.

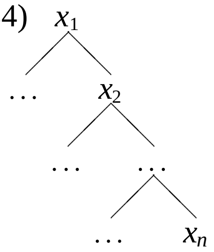
4.1 Modular effects

Modular effects in ambiguity resolution can arise in two ways in NL-Soar. First, NL-Soar may completely fail to detect an ambiguity, in which case knowledge cannot be brought to bear to resolve it. This is the most severe breakdown of ambiguity resolution possible, since the effects often cannot be overcome with additional knowledge or experience. The second kind of breakdown involves a failure to bring the required knowledge to bear on the ambiguity. Both kinds may give rise to apparent structural preferences in a variety of ways, as described in the next five sections.

4.1.1 The limit of two attachment sites

The strongest prediction that the A/R set makes about ambiguity resolution is that at most two nodes are available to assign the same structural relation at any given time. In a structure of the form

(84)



only two of the nodes will be available to assign any particular structural relation, even if all *n* sites are grammatically open. For example, consider the right branching structure in (85):

(85) Stewart saw the dog under the box on the table in the room next to the library.

At most two noun phrases are available to assign the adjunction relation (adjoin-N') to prepositional phrases.

Thus, the A/R set serves a theoretically similar function to *closure principles*, which predict when syntactic nodes are closed for further attachment. The best known are Early Closure (Kimball, 1973) and Late Closure (Frazier & Rayner, 1982) (§2.3.1). Church (1980) provides an empirical critique of both, demonstrating that Early Closure over-predicts difficulty and Late Closure under-predicts difficulty. He offers the *A-over-A Early Closure* principle as an alternative with significantly better coverage[1]. The critical idea is that the *two* most recent nodes of the same category (hence, the A over A) may be kept open at any time. This is similar to what the A/R set predicts, with the exception of the pure recency.

One way of directly testing the theory is to construct material with three potential sites and syntactically *force* attachment to each of the three sites as the experimental manipulation. NL-Soar predicts that one of the sites should cause difficulty, giving an impression of ungrammaticality. Recently, Gibson et al. (1993) conducted a study using material with three potential NP attachment sites, and found that forcing attachment to one of the sites (the intermediate site) caused difficulty. Although this study is not the best possible test of the theory[2], the same pattern of results held in an analysis of three-site NP-PP attachments in the Brown corpus: attachment to the intermediate site occurred just 14% of the time (Gibson & Pearlmutter, 1993).

The theory as stated does not predict which two of the three (or *n*) sites will be available, since it does not specify any particular strategy for determining which nodes remain in the

[1] Gibson (1991) uses a modified version of this principle (the *Principle of Preference Closure*).

[2] Because there was a momentary local ambiguity, an independent garden path effect may have been involved, which complicates the interpretation of the results.

A/R set and which nodes are replaced. One obvious possibility is to introduce an explicit recency preference, so that the two most recent nodes are held in the A/R set, but the data above suggests that this may not be correct, since the more recent intermediate site was significantly more difficult than the initial NP.

4.1.2 An emergent recency preference

Although the Gibson et al. (1993) study indicates that recency alone cannot account for the data, a general recency preference that can be modulated by other factors may still play an important role (Gibson, 1991). In fact, by abstracting away from the effects of any particular strategies for ambiguity resolution or handling conflicts in the A/R set, we can see that the basic structure of the A/R set does give rise to a kind of recency preference. More precisely,

(86) *A/R set recency preference:* Given a sequence of syntactic nodes $x_1, x_2, \ldots x_n$, $n > 2$, that potentially assign some structural relation ρ, attachment to more recent nodes via ρ is more likely than attachment to less recent nodes, all things being equal.

This preference can be derived with a simple probabilistic analysis. Let $P_S(x)$ be the probability that node x will be selected as an attachment site. Let $\prec$ be the precedence relation among nodes, such that $x \prec y$ means y is more recent than x. Then the general statement of recency preference is

$$\text{If } x \prec y \text{ then } P_S(x) < P_S(y) \tag{4.1}$$

Assume that $x_1, x_2, \ldots x_n$ denotes a sequence of syntactic nodes, so that if $i < j$, $x_i \prec x_j$.

Let $P_{WM}(x)$ be the probability that node x is in working memory indexed by some assigning relation ρ. Let $P_K(x)$ be the probability that search control knowledge selects node x in the A/R set for ρ-attachment. Then the probability P_S that a node will be selected as an attachment site is the probability that the node is in working memory *and* selected by search control:

$$P_S(x) = P_{WM}(x)P_K(x) \tag{4.2}$$

We abstract away from the effects of search control knowledge by assuming

$$\text{For all nodes } x,y\text{: } P_K(x) = P_K(y) \tag{4.3}$$

Each time an attempt is made to place a new node in the A/R set under some index ρ, there is a pool of three potential candidates: the two current members in the set, and the new potential member. Let $P_{AR}(x)$ be the probability that x is chosen to remain in the A/R set. We abstract away from strategies of maintaining the A/R set by assuming

$$\text{For all nodes } x,y\text{: } P_{AR}(x) = P_{AR}(y) \tag{4.4}$$

Of course, if the stream of syntactic nodes consists of one node x_1, then $P_{WM}(x_1) = 1$. Similarly for two nodes, $P_{WM}(x_1) = 1$, and $P_{WM}(x_2) = 1$. But at three nodes, $P_{WM}(x_i) = P_{AR}(x_i)$, for $i = 1, 2$ or 3. At four nodes, there are two opportunities to replace members of the A/R set, so we have

$$P_{WM}(x_i) = P_{AR}(x_i)P_{AR}(x_i) = P_{AR}(x_i)^2, \text{ for } i = 1, 2 \text{ or } 3; n = 4 \tag{4.5}$$

In general,

$$P_{WM}(x_i) = P_{AR}(x_i)^{n-i+1}, \text{ for } i, n > 2 \tag{4.6}$$

From 4.4 we have

$$\text{If } i < j \text{ and } i, j, n > 2, P_{AR}(x_i)^{n-i+1} < P_{AR}(x_j)^{n-j+1} \tag{4.7}$$

From 4.6 and 4.7 we have

$$\text{If } i < j \text{ and } i, j, n > 2, P_{WM}(x_i) < P_{WM}(x_j) \tag{4.8}$$

From 4.3 and 4.8 we have

$$\text{If } i < j \text{ and } i, j, n > 2, P_{WM}(x_i)P_K(x_i) < P_{WM}(x_j)P_K(x_j) \tag{4.9}$$

From 4.2 and 4.9 we have

$$\text{If } i < j \text{ and } i, j, n > 2, P_S(x_i) < P_S(x_j) \tag{4.10}$$

which is the recency preference (86).

This result only holds for nodes that are competing for the same structural index in the A/R set. Verbs do not compete with nouns for PP adjunction, nor do complement attachments compete with adjuncts. This is consistent with the fact that Right Association is not a good predictor across syntactic categories, or between argument/adjunct ambiguities (Abney, 1989). That is why the A-over-A Closure Principle discussed above is formulated in terms of nodes of the same category. This is borne out in the Whittemore & Ferrara (1990) study of PP attachments, where Right Association was found to be most effective in arbitrating noun-noun and verb-verb attachment ambiguities *not* accounted for by lexical argument structure.

To reiterate, this analysis neither assumes a recency preference, nor does it suggest one should be incorporated into NL-Soar. It is merely an attempt to reveal what effect the structure of the A/R set might have on ambiguity resolution *independent* of the particular strategies used to manage the contents of the set, or to perform ambiguity resolution itself. The demonstration shows that an apparent recency preference emerges as a basic property of the limited A/R set.

4.1.3 Object/subject and other “act vs. do-nothing” ambiguities

The kind of local ambiguities that emerge in parsing are a function of the particular parsing algorithm; different parsing schemes may exhibit different kinds of local ambiguities (Abney & Johnson, 1991). The head-driven, bottom-up process in NL-Soar sometimes shifts the detection of ambiguity to a point later than the earliest possible point that the ambiguity could be detected. This means that at the earlier point knowledge cannot be brought to bear to resolve the ambiguity.

Consider local object/subject ambiguities such as (87):

(87) Bob knows Susan went to the store.

The earliest possible point that the ambiguity may be detected is at *Susan*. However, detecting the ambiguity at this point would require positing the complement phrase for which *Susan* can serve as the subject. Since NL-Soar projects phrases from their heads, the complement phrase will not be created until *went* arrives—too late to affect the attachment choice of *Susan.*

The only ambiguity that exists at *Susan* is a choice between attaching *Susan* as the object, or doing nothing. Given NL-Soar’s control structure, such a choice is no choice at all. When one alternative is generated, the outcome of the decision procedure is to proceed with that alternative without further deliberation. Thus, NL-Soar exhibits a preference for objects in object/subject ambiguities. “Preference” is perhaps a misnomer since the system is not even considering the alternative.

The preference for objects in such ambiguous structures is well known in psycholinguistics (Hakes, 1972; Frazier & Rayner, 1982; Pritchett, 1992). The preference is generally detected in reading time studies, where subjects show an increased reading time in ambiguous sentences such as (87) over unambiguous controls. In the severe cases, the preference can even lead to a garden path effect (Chapter 6; Pritchett, 1992).

This kind of effect may arise in other structures as well. Consider the ambiguity in (88). *Green* will initially be taken as an NP complement, which turns out to be correct for (88a), but incorrect for (88b) (though no garden path effect arises; see Chapter 6).

(88) (a) I like green.
(b) I like green Martians.

It is possible to override this effect in NL-Soar, but it requires a deliberate attempt to force an impasse in the processing where none would otherwise exist. This can be accomplished by formulating an *explicit alternative* operator (perhaps a *do-nothing* operator), which would cause a tie impasse with the existing operator. Then search control associations might be learned that avoid taking the existing path. Such a scheme effectively places NL-Soar in a *careful comprehension mode,* in which each decision is re-evaluated. We will see shortly that this kind of processing might be useful in certain cases, but it cannot be the default way of performing real-time comprehension.

4.1.4 Time pressure: failure to bring knowledge to bear

At any given point in time, knowledge in Soar is distributed across the problem spaces in a variety of ways as a function of the experience of the system. The knowledge required to perform some ambiguity resolution may not be available in a form that permits real-time comprehension. Consider again the familiar main verb/reduced relative ambiguity:

(89) The car examined . . .

Ambiguities of this type can sometimes be resolved by appeal to semantics, specifically, whether or not the subject of the sentence is a plausible agent for the verb. In (89), it is more likely that the car was being examined rather than doing the examining. If a search control association exists that tests the appropriate semantic features and prefers the u-constructor corresponding to the reduced relative interpretation, then this knowledge can be brought to bear on-line during the comprehension process.

However, nothing guarantees that such an association will be available. If the knowledge is present only in the lower problem spaces, there may not be enough time to perform all the inferences necessary to make the right selection. (Recall that comprehension can proceed at an average rate of about four operators per word). Under press of time, there may be no alternative for the system but to select one interpretation randomly or by some default preference[3]. In any case, NL-Soar is behaving in a modular fashion since the required knowledge sources are not applied on-line.

NL-Soar makes predictions about the kind of ambiguities and contexts that will tend to create modular effects. The more specific the relevant knowledge sources, the more likely NL-Soar will fail to bring them to bear on-line, because specific knowledge sources are less likely to have been chunked. The more general the knowledge sources, the less likely modular effects will arise, since it is more likely that NL-Soar will have encountered the situations necessary to learn the required associations. Thus, we should expect interactive effects based on lexical semantics to be more pervasive than interaction with particular aspects of referential contexts.

4.1.5 Masking effects: linguistic *Einstellung*

The differential distribution of knowledge across problem spaces can give rise to another kind of effect that is independent of real-time issues. Sometimes knowledge in one space masks the presence of knowledge available in lower spaces, because the knowledge in the other spaces has not yet been chunked into a form that makes it immediately available in the higher space. If the immediately available knowledge is sufficient to proceed without impasse, then the system may never access the additional knowledge—because impasses provide the occasion to do so. This is known as the *masking effect* in artificial intelligence (Tambe & Rosenbloom, 1993), and it is fairly pervasive in Soar. For example, once Soar finds a particular sequence of moves that succeeds in a game or puzzle, it will tend to *always*

[3] The implemented system is not actually forced to respond to time pressure.

follow that sequence in future situations, regardless of whether the sequence is the most efficient. The knowledge to explore alternative paths may exist elsewhere in the system, but the learned associations continue to guide it down the path initially learned.

In psychology, this is known as *Einstellung* (Luchins, 1942): the application of a learned behavior in new situations in which the behavior is not necessarily useful. NL-Soar predicts that there is *linguistic* Einstellung, in which the application of aspects of linguistic skill may actually interfere with the functional demands of comprehension.

Linguistic Einstellung can arise in several ways in NL-Soar. The presence of already-learned u-constructors and s-constructors may mask the fact that an alternative interpretation exists, because the alternative corresponds to an operator that has not yet been chunked. This is easiest to see in the case of u-constructors. Suppose that the system is in a state such that a u-constructor exists (say, `u-constructor41`) to attach a subject to an incoming verb, but has not yet learned a u-constructor for the reduced relative construction. Then in sentences such as (89), the ambiguity will not even be detected when the verb arrives. At *examined*, the proposal association for `u-constructor41` will fire, and that u-constructor will be selected since there are no available alternatives. The occasion to learn the alternative construction must come via other linguistic input (perhaps unambiguous input), or else through some more deliberate means of forcing an impasse as discussed earlier.

Of course, it is unlikely that adult comprehenders would be missing u-constructors for any but the most rare syntactic constructions, or perhaps for constructions encountered in novel idiolects. S-constructors are more likely to be missing, since these are a function of the *semantics* of the linguistic input. More general s-constructors may mask interpretations that are more appropriate in some specific context. This would predict, for example, that people new to a particular task environment with its own task-specific sublanguage will initially bring to bear their existing skills (in the form of the existing applicable s-constructors) until the required interpretation operators are built up. Missing s-constructors can also have an effect on syntactic ambiguity resolution, since some downstream ambiguity may be resolved as a function of the semantic interpretation established so far.

Search control knowledge can also be masked. Consider again the main verb/reduced relative ambiguity in (89). Suppose that a search control association has been learned that encodes a general preference for the main verb reading. The general form of such as association is given in (90):

(90) *IF* `u-constructor41` is proposed (the main-verb construction)
THEN prefer `u-constructor41` as the best operator

Such a preference could have been learned in a situation where there was no semantic or pragmatic basis for making a choice, or in a situation where real-time demands did not permit access to the semantic/pragmatic knowledge.

In any future situations where `u-constructor41` is proposed, chunk (90) will fire, guiding NL-Soar down the main verb path even though there is knowledge in the lower spaces that might select an alternative interpretation. No impasse arises. The other control knowledge is masked because chunk (90) permits recognitional—though possibly incorrect—behavior to occur. Again, such behavior can be overcome with a deliberate attempt to reconsider each decision, but there is no way of knowing that the decision might need to be reconsidered until it is too late. This kind of masking of control knowledge is the paradigmatic way that Einstellung shows up in Soar.

4.2 Interactive effects

NL-Soar operates as an interactive architecture whenever search control associations that embody non-syntactic information guide the selection of u-constructors. Section 3.6.1 provided an example of syntactic ambiguity resolution based on referential context. Interactive effects can arise in principle because there are no architectural limitations on what may be encoded in the conditions of control associations. They are only limited by what is represented in working memory at the moment the ambiguity arises. In the cases discussed so far, we simply posited the appropriate search control association to effect the ambiguity resolution. A much stronger theory would explain the origin of these associations. The remainder of this section describes how NL-Soar can in fact learn such search control chunks.

As an example, we will use the main verb/reduced relative ambiguity, repeated below:

(91) The car examined . . .

Suppose that NL-Soar has already learned the u-constructors corresponding to the main verb and reduced relative constructions, and furthermore, has learned a general preference association for main verbs (90). The discussion of the masking effect above makes clear that overcoming this preference to learn the correct search control rule will be a nontrivial matter.

Figure 4.1 shows what happens on the initial pass through the sentence fragment. This figure introduces an alternative way of illustrating the system behavior that will prove more efficient for the current purposes. Each line beginning with a number and a single letter followed by a colon corresponds to a decision cycle. `O:` denotes an operator, `P:` denotes a problem space, and `I:` denotes an impasse (`I` and `P` not shown in this figure). Impasses will be indented to indicate processing in a subgoal. Individual chunk firings will sometimes be noted on separate lines; they do not correspond to separate decision cycles. This trace is generated from the actual system output.

In the initial pass, after the application of `s-constructor39`, both u-constructors (`u-constructor41`, main verb, and `u-constructor45`, reduced relative) are proposed in parallel, and the association fires that prefers the main verb u-constructor. As expected from the earlier discussion of masking effects, this results in the selection of `u-constructor41`. The s-constructor that follows creates the situation model objects representing the situation in which the car is doing the examining. NL-Soar has demonstrated a classic modularity effect.

Assume that the system has some capability to notice semantic anomalies, so that at some point the content of the situation model is annotated as implausible. A simple way to

```
READING WORD: the

  O: u-constructor34

READING WORD: car

  O: u-constructor36
  O: s-constructor39

READING WORD: examined

  Firing chunk-574               ; propose main verb
  Firing chunk-488               ; propose reduced relative
  Firing prefer-main-verb

  O: u-constructor41
  O: s-constructor44
```

FIGURE 4.1: First pass through *The car examined.* Upon encountering *examined*, two associations fire in parallel proposing operators corresponding to the main verb and relative clause structures. The main verb proposal immediately triggers a general association preferring this operator, so at the next decision cycle, the main verb operator is selected (`u-constructor41`).

accomplish this in the present example is with an operator that matches an inanimate agent and marks the situation model as implausible.

There are a number of possible ways the system could respond to such anomalies. One plausible response is to attempt to recomprehend the input more carefully. In reading, this could take place via regressive eye movements (Carpenter & Daneman, 1981; Just & Carpenter, 1987); in speech, by appealing to a short term verbatim memory, or simply asking the speaker to repeat. The details of how it happens are not important here. We simply assume that there is some deliberate attention that enables the system to comprehend the fragment again from the start. Furthermore, it is not critical *when* the decision is made to recomprehend. The present model decides to recomprehend as soon as the anomaly is detected. Other delay strategies are possible.

Figure 4.2 illustrates this process, starting with the last operator from Figure 4.1. Once the anomaly is detected, the *attend* operator is selected, representing the intention to reprocess the input more carefully. What does it mean for NL-Soar to comprehend more *carefully*? The answer to this question was sketched in the §4.1.3. Decisions that were previously made without impasse must now be reconsidered.

When NL-Soar comprehends carefully, it forces impasses in situations where there is more than one operator proposed. The second time through, an impasse arises at *examined*, permitting more thorough evaluation of each alternative. The u-constructors are

```
  O: notice-anomaly
  O: attend

READING WORD: the

  O: u-constructor34

READING WORD: car

  O u-constructor36
  O s-constructor39

READING WORD: examined

  Firing chunk-574                ; propose main verb
  Firing chunk-488                ; propose reduced relative
  Firing prefer-main-verb

  ==>I: operator tie
     P: Selection
     O: evaluate (u-constructor41)
     ==>I: operator application
        P: Comprehension
        O: u-constructor41
        O: s-constructor44
        O: evaluate-situation-model

  Evaluation of u-constructor41 is implausible

     O: evaluate(u-constructor45)
     ==>I: operator application
        P: Comprehension
        O: u-constructor45
        O: s-constructor53
        O: evaluate-situation-model

  Evaluation of u-constructor45 is plausible
  Build: chunk-597

 O u-constructor45              ; reduced relative
 O s-constructor53
```

FIGURE 4.2: Carefully re-comprehending *The car examined.* An impasse is forced at the point where the two operators are proposed, and the decision to select the main verb operator is reconsidered. Each operator is evaluated by applying it and evaluating the resulting situation model for plausibility. A preference is returned for the relative clause operator, creating a new chunk.

evaluated by simply applying them, allowing the s-constructors to perform the semantic interpretation, and then comparing the interpretations. `U-constructor41` produces an implausible evaluation as it did earlier; `u-constructor45` produces a plausible evaluation. Based on these two evaluations, a preference is returned preferring `u-constructor45` over `u-constructor41`. This resolves the original impasse, resulting in the selection of `u-constructor45`, and processing continues in the top space as usual.

The resolution of the impasse produces a search control chunk (`chunk-597`) of the form:

IF `u-constructor41` and `u-constructor45` are proposed
and the incoming verb is *examine*
and the preceding NP refers to something inanimate
THEN prefer `u-constructor45` over `u-constructor41`

This is precisely the kind of semantic association we assumed could exist in principle in the earlier discussions of interactive and modular ambiguity resolution. Critically, it is not conditioned upon NL-Soar being in careful comprehension mode, since that was irrelevant to the problem solving that produced the chunk (of course, it was exceedingly relevant to *initiating* the problem solving)[4].

Figure 4.3 shows what happens now that the chunk is in long term memory. At the ambiguity, the two u-constructors are proposed as usual. This is immediately followed by the firing of the two control associations: the general preference for `u-constructor41`, and the semantic preference for `u-constructor45` over `u-constructor41`. `U-constructor45` is then selected without further deliberation (because more specific binary preferences take precedence over unary preferences in Soar). NL-Soar now exhibits classic interactive effects.

This scheme is an instantiation of the general method of *recovery from incorrect knowledge* in Soar (Laird, 1988). The distinguishing feature of this recovery in Soar is that the *incorrect decision* is corrected by monotonic additions to the long-term memory—no association is changed or removed. The original general preference does not go away with the acquisition of the new semantic control association. This is a necessary result of the fundamental assumption in Soar that long term memory is permanent and cognitively impenetrable. The deliberate nature of the method (the requirement to force impasses) is a direct result of the masking effect noted earlier.

Although recomprehending is a very simple scheme for error recovery, it has a number of features that make it fairly plausible. There is abundant evidence from eye movement studies for regressions during comprehension, with pauses at difficult material (Just & Carpenter, 1987). The simplicity of the scheme has functional advantages as well. There is no credit assignment problem: NL-Soar did not know for certain in advance that the ambiguity was the source of the problem, nor did it *need* to know. In fact, it didn't even know for certain that a miscomprehension was the source of the ambiguity. There is no guarantee that this procedure will yield the desired result.

[4] This chunk tests for the specific lexical item; see §3.4.3 for a discussion of how this is a function of what semantic features are retrieved by lexical access.

```
READING WORD: the

  O: u-constructor34

READING WORD: car

  O: u-constructor36
  O: s-constructor39

READING WORD: examined

  Firing chunk-574                ; propose main verb
  Firing chunk-488                ; propose reduced relative
  Firing prefer-main-verb
  Firing chunk-597                ; prefer relative to main verb

  O u-constructor45                    ; reduced relative
  O s-constructor53
```

FIGURE 4.3: Comprehending *The car examined* after learning. The new semantic search control chunk-597 fires, guiding the syntactic parsing down the correct path.

4.3 Summary: The NL-Soar theory of ambiguity resolution

The theory of ambiguity resolution described above has three components: a set of functional mechanisms designed to approximate a knowledge-level view of ambiguity resolution, a variety of ways that these mechanisms fail to reach the knowledge level, and the means by which some of these limitations can be overcome with learning.

NL-Soar is first and foremost a *functional* theory of language comprehension, just as Soar is primarily a functional theory of cognitive architecture. The model thus embraces what might be called the *knowledge-level* theory of ambiguity resolution:

> *Knowledge-level theory of ambiguity resolution:* Any knowledge source may be brought to bear to resolve local ambiguities in language comprehension.

Such a theory places no limits on the kinds of knowledge that can affect ambiguity resolution: the knowledge can range from simple semantic restrictions on verb arguments to details of the current problem solving context. This knowledge must be specified to make predictions of behavior. The referential theory of Altmann, Crain, and Steedman (§2.3.5) is a prime example of a knowledge level theory of ambiguity resolution.

NL-Soar supports knowledge-level ambiguity resolution with Soar's open control structure (§3.6). There are no architectural limits on the knowledge encoded in the search control associations that guide processing, and these knowledge sources are brought to bear on-line at each decision cycle. We have seen examples of NL-Soar using referential context (§3.6.1) and semantic content (above) to resolve ambiguities on-line.

However, we have also seen that the symbol-level mechanisms that constitute NL-Soar may fail to perfectly realize the knowledge level, and this failure can be detected behaviorally[5]. The following list summarizes the ways this may happen:

1. Only immediately available knowledge can affect resolution on-line (under press of time); what is immediately available is a function of experience.

2. The A/R set permits only a small subset of the syntactically permissible attachments to be detected; this also leads to a general recency preference.

3. Immediately available search control knowledge may mask other search control knowledge sources present in the system.

4. Immediately available comprehension operators may mask other possible semantic or syntactic alternatives.

5. Some ambiguities (e.g., subject/object) are not detected immediately because the alternatives emerge late in head-driven processing.

The final component of the theory is the role of learning and experience in ambiguity resolution. The example in §4.2 shows how the symbol-level failures can be overcome with more careful reprocessing of the linguistic input, and how this reprocessing gives rise to new associations that effect knowledge-level ambiguity resolution on-line. Thus, NL-Soar not only explains how interactive and modular effects arise, but it also provides the mechanisms by which the shift can be made from modular to interactive behavior.

4.4 Is NL-Soar modular?

Now that the theory of ambiguity resolution has been described in some detail, it is instructive to step back and ask: Is NL-Soar modular?

Considering the structural relationship to central cognition, NL-Soar is clearly *nonmodular,* because NL-Soar uses exactly the same set of mechanisms that underlie all cognition in Soar. No new architectural features were posited in NL-Soar to comprehend language.

However, a somewhat different view is obtained by considering the distribution of knowledge in the system across comprehension operator functions. Syntactic knowledge is contained in the proposal and applications of u-constructors. Furthermore, the proposal and application of u-constructors consists exclusively of syntactic knowledge. (Recall from

[5] The imperfect relationship between symbol level and knowledge level is the existential bane of physical computational devices, and is also the means by which psychological theorizing about architecture can take place (Newell, 1982; Newell, 1990).

§3.7 that this basic distribution of knowledge sources was motivated by computational and functional concerns). Semantic interpretation knowledge is held in the s-constructors; semantic and contextual search control knowledge is held in the search control associations for comprehension operators. Thus, the large set of associations that propose and apply u-constructors in effect comprises an informationally encapsulated syntactic module.

We can gain further insight into the issue by considering NL-Soar along several of Fodor's (1983) dimensions for characterizing modularity. Newell (1990) did this in the context of the initial Soar comprehension theory, but his analysis still holds. What follows is a partial summary of this analysis with respect to the present NL-Soar.

Domain specificity. Modules become highly tuned to the specific characteristics of the domain (in Fodor's analysis, the possible domains are perception (vision, audition. etc.), plus language). Chunking is precisely a system for building up mechanisms (chunks) tuned to specific aspects of the environment. The chunks that comprise NL-Soar's recognitional comprehension capability form a special-purpose system. As Newell pointed out, "In Soar, the generality of cognition is not that every situation must be treated generally, but that generality is always possible when knowledge is missing (impasses) and that the construction of special-purpose mechanisms (chunks) works to avoid future breakout."

Mandatory operation. The linguistic module applies in a mandatory fashion: one must treat received speech as speech. The recognition memory in Soar is also mandatory and automatic. For example, comprehension operators are proposed automatically whenever their relevant inputs are present in working memory. Once the process has been chunked, there is no way to avoid this. The decision cycle does provide the level of control that may permit something different to be done with the input. But as we have seen, the masking effect in Soar mitigates heavily in favor of whatever recognitional skill can be applied at the moment. Special, deliberate modes of processing along with sufficient practice are required to build up the skills that might compete as alternatives to the existing comprehension operators. Even then, the comprehension operators will still be applicable, so the emerging behavior will be a mix of the existing and new skills. Furthermore, the encoding productions (assumed to handle much of lexical access in NL-Soar) fire automatically without any control from central cognition, so that only post-lexical-access processing can be modulated in this way.

Fast operation. Language comprehension is a real-time, rapid process that presses the limits of neural technology. The basic structure of NL-Soar is fundamentally responsive to this constraint. NL-Soar comprehends each word with a few 50 ms operators per word. The entire structure of the recognition memory in Soar is designed to bring knowledge to bear rapidly. Chapter 7 will deal with issues of real-time immediacy of interpretation in greater detail.

Information encapsulation. Modules access a limited set of distinguished knowledge sources; they are not open to the general inferences of central cognition. As we have seen, the chunks comprising the proposal and application of u-constructors access only syntactic knowledge. In general, the knowledge brought to bear by the recognitional comprehension capability is limited by whatever is encoded at the moment in the chunks that implement the top-level comprehension operators. But we have also seen that it is possible to penetrate this capability with the results of general cognitive activity. As Newell points out, however, "whether Soar would ever be overwhelmed by a loss of control in a great rush of cogitations seems dubious." Arbitrary processing only comes in to play when impasses arise. Given a highly-practiced skill such as comprehension, and given the ubiquitous masking effect in Soar, the frequency of such impasses will be rather limited. And even during the resolution of an impasse, control is not lost. Soar does not have a subroutine control structure—any processing is interruptible at the next decision cycle.

We arrive (somewhat appropriately for this chapter) at an ambiguity. NL-Soar can be seen as having many of the critical characteristics of modular systems, both structurally and behaviorally. Yet, it also has many of the characteristics of an interactive system, both structurally and behaviorally.

NL-Soar essentially provides the same fundamental answer as the modularity thesis to the question of why certain limitations in ambiguity resolution arise: the limits derive primarily from a system structured and tuned to perform comprehension in real-time. However, the route by which NL-Soar arrives at this answer—an approach concerned primarily with functionality and positing specific mechanisms, and embedded in a general cognitive architecture—has yielded a much richer theory than might otherwise have been possible. NL-Soar explains modular and interactive behavior on the basis of more general principles of cognitive architecture. It predicts that limitations will flow from *functionally* motivated aspects of the model (Young, 1993). It explains how the system, through experience, might overcome these limitations and make the shift from modular to interactive behavior. Finally, it addresses in a deep way the relationship between automatic and deliberate processes, opening the door to a better understanding of the relationship of language and cognition generally, rather than leaving central cognition as an unexplicated black hole (Lewis, 1992; Lehman et al., 1993b).

4.5 General discussion

This chapter has painted a fairly complex picture of ambiguity resolution, but it all emerges from a small set of assumptions: the basic control structure and learning mechanisms of Soar, plus the limited structure of the A/R set. The predictions are consistent with what is known about ambiguity resolution, which itself paints a fairly complex picture.

One potential problem for a theory of ambiguity resolution such as the one presented here is the difficulty of falsification. If ambiguity resolution is guided by any immediately available knowledge source, then potentially any result can be accounted for by positing the right knowledge sources. There is a genuine danger here. However, there is a resolution to this difficulty. Ultimately, NL-Soar must not be just a theory that specifies how multiple knowledge sources interact, but a theory of the *acquisition* of that knowledge as well. Section 4.2 sketched the beginnings of just such a theory. With an acquisition model, the relevant associations that accomplish ambiguity resolution are not posited by the theorist. Instead, they arise when the model is placed in particular learning situations. New experiments might be devised that explicitly test the theory by placing subjects in *linguistic* *training situations,* and using the traditional pre- and post-tests to determine the behavioral changes that result from learning.

In any event, it is certainly the case that NL-Soar is able to make predictions concerning ambiguity resolution independently of posited immediate knowledge sources. Section 4.1 provided several examples: the limited subset of attachment sites, the general recency preference, the preference for objects over subjects. On this score, NL-Soar has proven accurate. As mentioned in Chapter 2, the challenge for theories that make clear predictions of structural biases in every ambiguous situation is accounting for the plasticity of resolution demonstrated across a range of contexts and ambiguity types. For these effects, NL-Soar provides a consistent account, and pushes the state of the science farther than any existing model by beginning to show how these interactive knowledge sources can be arise.

Chapter 5

Parsing Breakdown and Acceptable Embeddings

PARSING BREAKDOWN ON CENTER-EMBEDDINGS is one of the best known phenomena in cognitive psychology, and the number of theories proposed over the years to explain it attests to this fact (§2.5). However, it was not until Gibson (1991) that any theory dealt with the variety of difficult embeddings, and perhaps even more importantly, the variety of complementary *acceptable* embeddings. This chapter describes NL-Soar's account of both difficult and acceptable embeddings. The first section outlines the theory of parsing breakdown, derived directly from the structure of the model presented in Chapter 3, particularly the A/R set. Then the theory is applied in detail to the 43-sentence collection of difficult and acceptable embeddings presented in Chapter 2. Next we consider how NL-Soar accounts for the major qualitative phenomena surrounding parsing breakdown. The chapter concludes with a brief discussion and summary of the results. Since the A/R set plays a role in explaining all the major phenomena addressed in this thesis, a full discussion of the A/R set and the "magic number two" will be delayed until Chapter 9.

5.1 The NL-Soar theory of parsing breakdown

NL-Soar's theory of parsing breakdown belongs to the class of *architectural theories* of breakdown (§2.5), rather than the class of structural metrics (of course, a metric can be derived from any architectural theory, but the converse is not necessarily true). The predictions of the theory primarily derive from the A/R set, with its two-valued syntactic indices.

Recall again the structure of the A/R set (§3.3.3). This is the data structure in working memory that indexes nodes in the utterance model by their potential syntactic relations. The set of syntactic relations corresponds to X-bar structural positions (spec-IP, comp-V', etc.). Each relation indexes no more than two nodes. Thus, parsing breakdown will occur whenever a particular syntactic structure requires that a relation index three or more nodes.

More precisely, breakdown occurs at the point when a node is needed in the parse but it is not available in the A/R set. Consider the classic center-embedded object relative[1]:

(93; PB1) The man that the dog that the boy saw liked ran away.

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

RECEIVERS	spec-IP:	[$_{NP}$ *the man*],[$_{NP}$ *the dog*],[$_{NP}$ *the boy*]

Breakdown will occur at one of the final verbs (which verb depends on which NP is dropped from the A/R set). Breakdown does *not* occur simply as a result of nodes dropping from the set. In fact, this happens continuously, without leading to unacceptability. The clearest example is the classic right branching structure:

(94; AE1) The dog saw the man that chased the boy into the table that the cop hit.

In (94), the stream of NPs clearly overloads the A/R set (particularly, the adjoin-N' relation in the assigners set), but no breakdown occurs because no more than one NP must be held at any given time. (Recall the empirical evidence presented in Chapter 4 that suggests only a subset of nodes in a right branching structures are available for attachment.)

This is essentially an *interference* theory of short-term memory for syntactic structure. The capacity of the A/R set is not defined absolutely in terms of number of syntactic nodes, but rather is a function of the *syntactic content* of what is to be stored. When nodes must be indexed by the same syntactic relations, they interfere with each other, and the fixed capacity is quickly reached. Nodes indexed by different relations, however, do not press the limits of the structure. Thus, the total capacity of the A/R set is a function of the set of available syntactic discriminators (assumed here to be X-bar structural positions).

The distinguishing characteristics of this theory can be clarified by comparing it with other models of short-term linguistic memory. The theory differs from *content-independent* theories of storage limitations, such as Yngve's (1960) original stack-based model, which posits a fixed-capacity stack used for the uniform storage of syntactic nodes. NL-Soar's

[1] A few examples have been given in the literature of doubly center-embedded object relatives which appear to be more acceptable. They include:

(92) (a) The guy whom the secretary we fired slept with is a real lucky dog. (Kimball, 1975)

(b) Isn't it true that example sentences that people that you know produce are more likely to be accepted? (De Roeck et al., 1982)

(c) A syntax book that some Italian that I had never heard of wrote was published by MIT press. (Frank, 1992)

Although such sentences have been proposed as examples of semantic or pragmatic effects, one striking commonality appears to be the use of a pronoun in the most deeply embedded clause (*I*, *we*, and *you* above). Given this fact, it seems premature to classify these examples as purely semantic or pragmatic effects. Drawing the correct generalization and accounting for the effect within the current theory will be an area for future research.

model has a *content-dependent* capacity. The theory differs from *semantic forgetting* theories of syntactic structure, which posit that syntactic structure is removed from short-term memory as it becomes semantically interpreted (Frazier, 1985). In NL-Soar, syntactic structure drops from short-term memory purely as a function of the incoming syntactic structure, regardless of what other processing has been done on it. The theory also differs from *uniform resource* theories, such as CC READER, which posits a single computational resource shared across all contents of working memory.

5.2 Predictions on the PB and AE collection

This section describes in detail NL-Soar's predictions on the collection of 43 parsing breakdown constructions and acceptable embeddings (Tables 2.11–2.15). The predictions are derived as follows: if a construction requires three or more nodes to be indexed under the same structural relation, then the construction is predicted to cause parsing breakdown; otherwise, the construction is predicted to be acceptable. This is the simplest method of applying the theory since it abstracts away from any specific strategies for handling conflicts in the A/R set (i.e., how to choose which nodes remain in the A/R set when three or more are vying for the same indexical relation). This method cannot overpredict difficulty because no strategy for maintaining the A/R set can overcome the two node limitation. However, it may overpredict acceptability, in that there exist A/R set maintenance strategies that do not ensure that the appropriate nodes are available for attachment. For example, one rather perverse strategy is to admit the first two nodes under each index and then block all subsequent nodes. Such a strategy would be completely disfunctional, of course. We will return to the issue of A/R set maintenance later in the chapter.

As in the example above, the predictions will be illustrated by presenting a partial view of the A/R set at critical moments. For structures causing parsing breakdown, the A/R set will be presented at the point where one of the relations must bear three values for the comprehension to continue successfully. For acceptable structures, the A/R set will be presented at points where the A/R set bears its maximum load. The phrase structure tree will often be given representing the final or intermediate output of NL-Soar. These trees are generated automatically from a trace of NL-Soar's working memory and edited for brevity.

The predictions are grouped into aggregations of similar structures, with both acceptable and breakdown constructions considered in each group. A summary table of the results appears at the end of the chapter.

5.2.1 Right and left-branching

Large amounts of right branching (94) are acceptable as indicated above, since typically just one node (the most recent) must be available for attachment. But there is more to the story. Purely bottom-up parsers encounter difficulty with right branching (Chomsky & Miller, 1963), so why doesn't the head driven, bottom-up strategy of NL-Soar? The problem is with traditional bottom-up parsers that expand phrase structure rules similar to

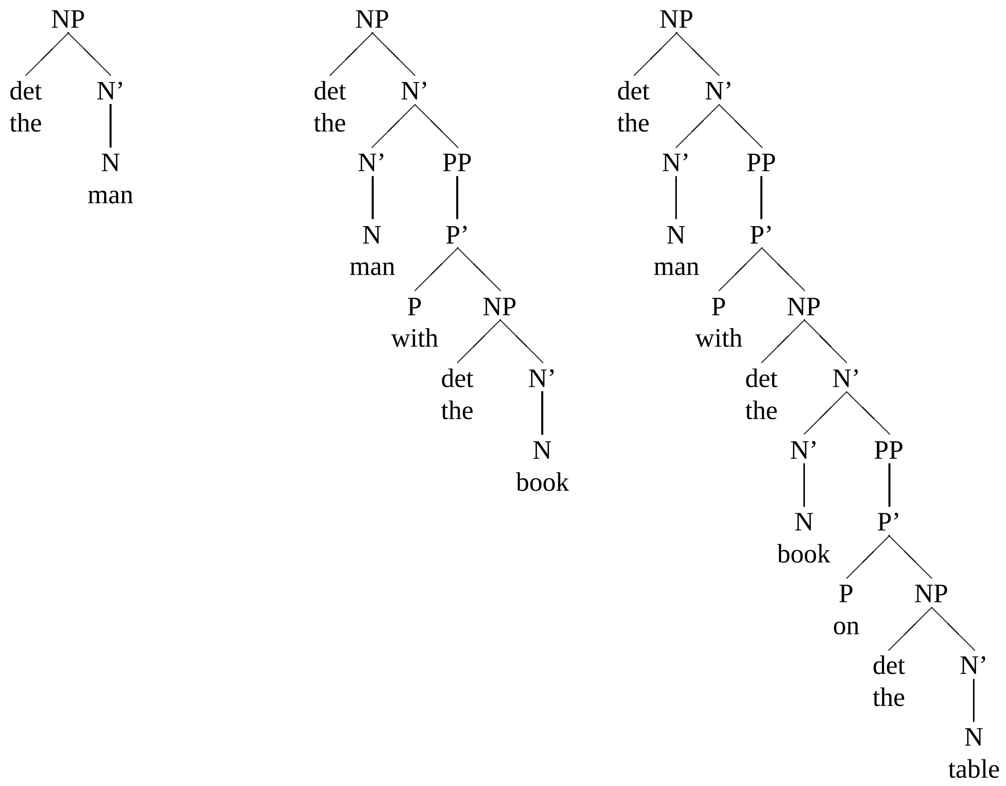


FIGURE 5.1: Growth of right branching structures.

NP → NP PP
NP → NP S'

With a right branching structure, the parser must keep open every node for possible further bottom-up reduction. For example, in *John saw the book on the table*, the NP *the table* cannot be removed from consideration for further expansion, since the sentence may continue *in the room*. This in turns means that the preposition *on* must remain in consideration, since it may need to attach to a new NP. Similarly, *the book* may need to attach to a new PP, and so on—unbounded right branching structures lead to unbounded requirements for buffering partial constituents.

NL-Soar avoids this because it is not choosing from a set of phrase structure rules to expand. A modified NP may be formed by simply adjoining to an existing NP. Figure 5.1 shows part of the incremental growth of the utterance model for (94). There is no need to delay attachment decisions. The only limit that can arise is in the number of potential attachment sites, an issue addressed above and in Chapter 4.

Left branching structures are also acceptable (Figure 5.2):

(95; AE2) Her sons' employees' contributions . . .

Because NL-Soar is building the tree bottom-up, left branching structures are easily handled (Chomsky & Miller, 1963).

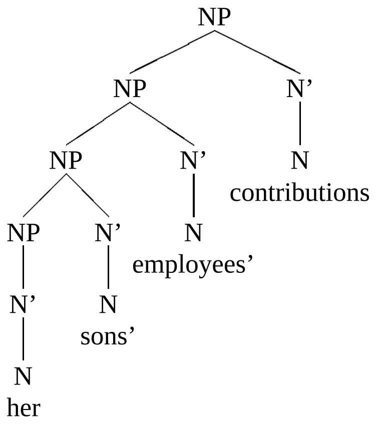


FIGURE 5.2: Left branching.

5.2.2 Center-embedded relatives

A single center-embedded relative clause can be parsed because just two NPs must be buffered in the spec-IP (subject) relation of the receiver's set (phrase structure in Figure 5.3).

(96; AE3) The dog that the cop saw is big.

RECEIVERS	spec-IP:	[$_{NP}$ *the cop*],[$_{NP}$ *the dog*]

As we saw earlier, two such embedded relatives (PB1) leads to breakdown because three NPs must be available on the spec-IP relation. It is irrelevant whether the overt complementizers are present or not, both structures are predicted to cause difficulty:

(97; PB2) The man the woman the dog bit likes eats fish.

An object relative may be embedded in a Wh-question without causing breakdown:

(98; AE4) What did the man that Mary likes eat?

In (98), the NP [$_{NP}$ *what*] is attached immediately in spec-CP position, as in Figure 5.4, so does not contribute to interference on the spec-IP relation. In fact, [$_{NP}$ *the man*] is also attached immediately, occupying spec-IP position of the IP projected from *did*.

Now consider a difficulty embedding in in a Wh-question (Gibson, 1991):

(99; PB3) Who did John donate the furniture that the repairman that the dog bit found to?

An overloaded spec-IP relation on the receivers set cannot be the source of difficulty in (99). [$_{NP}$ *What*] and [$_{NP}$ *John*] need not be buffered in the receivers set because they are attached immediately as in Figure 5.4. Surprisingly, however, NL-Soar does predict difficulty here for another reason. Consider the relation of object traces and antecedents in (99):

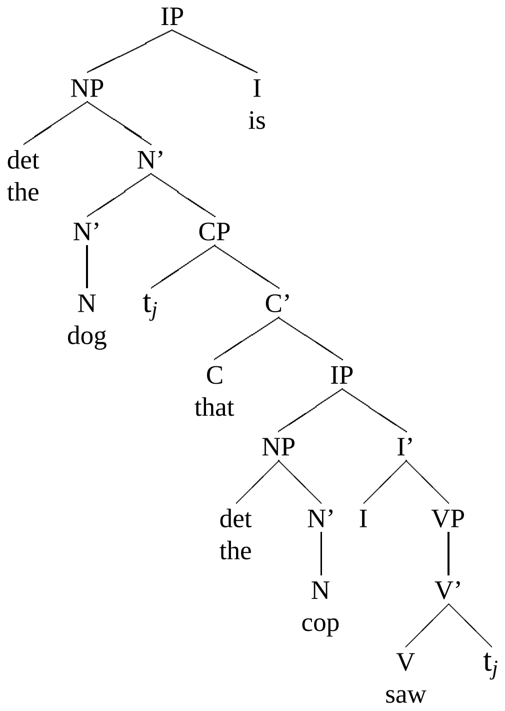


FIGURE 5.3: Singly-embedded object relative.

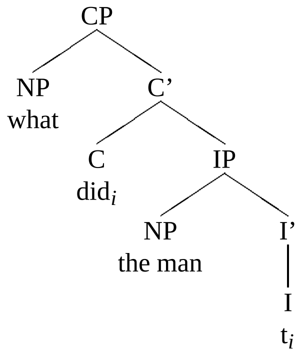


FIGURE 5.4: Structure for Wh-question.

(100) Who$_i$ did John donate the furniture$_j$ that the repairman$_k$ that the dog bit t$_k$ found t$_j$ to t$_i$?

Each of the traces is generated by accessing the antecedents in spec-CP position via the assigners set. By the second relative clause, three CP nodes must be indexed on the spec-CP assigners relation in order to generate the correct traces:

ASSIGNERS	spec-CP:	[$_{CP}$ *who did*],[$_{CP}$ *that*], [$_{CP}$ *that*]

Thus, (99) causes difficulty because of an overload on the *assigners* set, rather than the receivers set.

Eady and Fodor (1981) discovered that placing the modified NPs in post-verbal position increases the acceptability of center-embedded relatives ((103) from (Gibson, 1991)):

(101; AE5) The cop saw the man that the woman that won the race likes.

(102; AE6) The cop saw the man that the boy that the dog bit likes.

(103; AE7) John donated the furniture that the repairman that the dog bit found in the basement to charity.

Nl-Soar predicts that these structures are acceptable because once [$_{NP}$ *the man*] is attached in complement position, it is no longer indexed in the receivers set. In the case of the object relative (102), the A/R set must index at most two NPs on the spec-IP relation:

RECEIVERS	spec-IP:	[$_{NP}$ *the boy*],[$_{NP}$ *the dog*]

Note that the creation of the trace in object position does not require the antecedent to be in the receivers set. The antecedent is accessed via the assigners set in spec-CP position (§3.3.3).

In the subject-relative (101), at most *one* NP must be indexed in the receivers set ([$_{NP}$ *the woman*]). This means that center-embedded subject-relatives are predicted to be acceptable, even in preverbal position, in contrast to the difficult object-relatives:

(104; PB4) The man that the woman that won the race likes eats fish.

Unlike (93) or (97), only two NPs ([$_{NP}$ *man*] and [$_{NP}$ *woman*] must be indexed simultaneously. Holding to the judgments presented in (Gibson, 1991), this is the first incorrect prediction of the model. However, at least as far as *relative* difficulty is concerned, this prediction is in the right direction. There is evidence that object relatives are more difficult in general than subject relatives (Ford, 1983; Holmes & O'Regan, 1981). Unfortunately, all the empirical evidence on double center-embeddings uses object relatives, not subject relatives, so for now informal linguistic judgments for (104) must suffice.

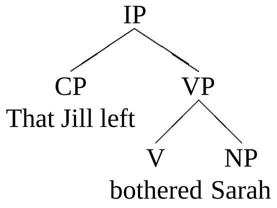


FIGURE 5.5: The wrong way to analyse subject sentences.

5.2.3 Subject sentences

Subject sentences are sentences that appear as subjects of other sentences:

(105; AE8) That Jill left bothered Sarah.

Kimball (1973) noted the unacceptability of embedded subject sentences:

(106; PB5) That that Jill left bothered Sarah surprised Max.

The most straightforward analysis of subject sentences places the complementized sentence (CP) directly in subject (spec-IP) position (Rosenbaum, 1967; Gibson, 1991), as in Figure 5.5. Under such an analysis, NL-Soar would predict the difficulty of (106), since three nodes must occupy the spec-IP relation:

RECEIVERS	spec-IP:	[$_{CP}$ *that*],[$_{CP}$ *that*], [$_{NP}$ *Jill*]

However, Koster (1978) presents compelling evidence that this analysis is incorrect. He points out a number of anomalies that arise as a consequence, including the two below:

(107) (a) *Although that the house is empty may depress you, it pleases me.
(b) Although it may depress you that the house is empty, it pleases me.

(108) (a) *Did that John showed up please you?
(b) Did John's showing up please you?
(c) *What does that he will come prove?
(d) What does his coming prove?

Subject sentences are generally ungrammatical in subordinate clauses (107), and cannot serve as the subject in subject-auxiliary inversions (108). But if the structure in Figure 5.5 is correct, these constructions should be acceptable.

Koster provides an alternative analysis that places the subject sentence in a *topicalized* position bound by a phonologically null trace. Such structures are already required to handle topicalization in English (Koster presents additional Dutch examples):

(109) (a) Clever$_i$ she certainly is t$_i$.
(b) This book$_i$, I asked Bill to read t$_i$.

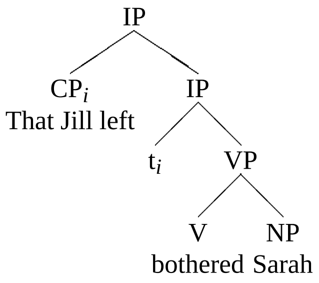


FIGURE 5.6: Topicalization analysis of sentence subjects.

Since topicalization is a *root* phenomenon (i.e., it generally occurs only at the level of the main clause, not in embedded contexts), this analysis predicts the unacceptability of (106) on *grammatical* grounds, because (106) requires an embedded topicalization. A relative clause with an embedded subject sentence should also be unacceptable:

(110; PB6) *The woman that for John to smoke would annoy works in this office.

(111; PB7) *The company hired the woman that for John to smoke would annoy.

Koster's 1981 treatment of topicalization involves a new phrase structure rule. In modern GB syntax topicalization is simply analysed as adjunction to IP (Rochemont & Culicover, 1990). Figure 5.6 shows the revised structure for subject sentences.

This analysis has immediate processing implications. Since the subject sentence no longer occupies the spec-IP position, is should be possible to embed a subject NP modified by an object relative clause *within* a subject sentence:

(112; AE9) That the food that John ordered tasted good pleased him.

This is correctly predicted to be acceptable, since the three initial phrases are distributed across two structural relations in the A/R set:

RECEIVERS	adjoin-IP:	[$_{CP}$ *that*]
	spec-IP:	[$_{NP}$ *the food*], [$_{NP}$ *John*]

Since fronted clauses are also adjoined to IP, a fronted clause followed by a subject relative is acceptable:

(113; AE10) While Mary slept, the sock that the dog chewed fell on the floor.

The same prediction holds for topicalized NPs as well:

(114; AE11) Bob, the girl that the dog scared likes.

By nominalizing subject sentences, it is possible to embed them without causing parsing breakdown (Kimball, 1973; Gibson, 1991):

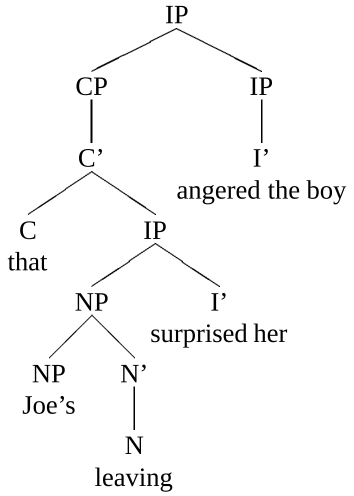


FIGURE 5.7: Nominalized subject sentences.

(115; AE12) That Joe's leaving surprised her angered the boy.

The structure for (115) is shown in Figure 5.7. NL-Soar handles such structures because they are left branching.

Although the restriction on embedded topicalization appears to be universal, Green (1976), Hooper and Thompson (1973), and Koster (1978) himself have all noted that root phenomena are sometimes marginally acceptable in the complements of a restricted class of English verbs:

(116) (a) I know that John, she likes.
(b) *I persuaded him that Bill, Herb likes.
(c) *I hoped that John, she likes.

Thus, in some cases embedded sentential subjects will be acceptable[2]:

(118; AE13) The cop believes that for the boy to smoke is bad.

(119; AE14) Mary held the belief that for John to smoke would be annoying.

Although these constructions may be grammatically marginal, the question for NL-Soar is whether such structures can be processed. They can in fact be processed; no relation must index more than two nodes. Figure 5.8 gives the structure produced for (118).

[2] The tensed versions of these subject sentences appear to be far less acceptable:

(117) (a) ?*I believe that that John smokes annoys me.
(b) ?*Mary held the belief that that John smokes is annoying.

It seems that there is some problem related to the repeated *that*'s which is independent of any of the issues presented here, but it is entirely unclear whether the problem is a processing issue or a grammatical issue.

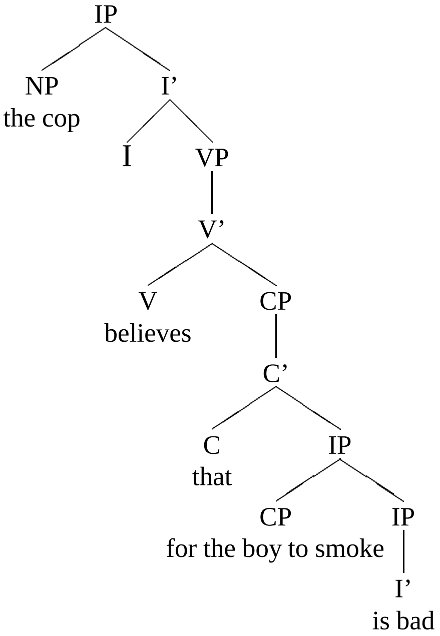


FIGURE 5.8: Acceptable embedded subject sentences.

This does lead to a problematic prediction: a subject sentence embedded in a complement of a subject NP should be acceptable, but in fact it does seem to cause some difficulty (Gibson, 1991):

(120; PB8) ?Mary's belief that for John to smoke would be annoying is apparent due to her expression.

NL-Soar can process this precisely because the subject sentence can be indexed via the adjoin-IP relation:

RECEIVERS	adjoin-IP:	[$_{CP}$ *for*]
	spec-IP:	[$_{NP}$ *belief*], [$_{NP}$ *John*]

However, the contrast between (120) and (119) is not particularly striking. The marginal grammatical status of embedded subject sentences in general makes it a somewhat difficult to evaluate processing theories against these constructions.

5.2.4 Complements of nominals

Using nouns that take sentential complements (*the belief that*, *the possibility that*), it is possible to create constructions that require buffering three or more subject NPs without using any relative clause modification. NL-Soar correctly predicts the difficulty on such embeddings:

(121; PB9) John's suspicion that a rumor that the election had not been run fairly was true motivated him to investigate further.

RECEIVERS	spec-IP:	[$_{NP}$ *suspicion*], [$_{NP}$ *rumor*], [$_{NP}$ *election*]

The same prediction holds for certain mixes of complement clauses and relative clauses:

(122; PB10) The man who the possibility that students are dangerous frightens is nice.

By using a subject relative, it is possible to create an acceptable embedding of a relative clause within a complement cause:

(123; AE15) The thought that the cop that hit the boy was rich angered her.

NL-Soar handles this because *thought* and *cop* are the only NPs that must be indexed on the spec-IP relation. Figure 5.9 shows the phrase structure.

RECEIVERS	spec-IP:	[$_{NP}$ *the thought*], [$_{NP}$ *the cop*]

An object relative embedded within an NP complement should also cause breakdown, but the result is considerably more acceptable than (122):

(124; AE16) The thought that the man that John liked screamed scared me.

This is an incorrect prediction by NL-Soar, but, as in the earlier case, it does at least correctly predict that object relatives are more difficult than subject relatives.

NL-Soar correctly predicts that NPs with sentential complements may appear in subject-auxiliary inversions:

(125; AE17) Who did the information that Iraq invaded Kuwait affect most?

However, NL-Soar also predicts that another embedded clause should be acceptable:

(126; PB11) Who does the information that the weapons that the government built don't work properly affect?

This incorrect prediction arises because *does* projects both CP and IP phrases, so that *information* is attached in spec-IP position as soon it is encountered (see again Figure 5.4). Thus, only *weapons* and *government* must be indexed in the A/R set simultaneously.

As with the case for embedded relatives, moving the complement embeddings to post-verbal position increases their acceptability:

(127; AE18) The pentagon employs many bureaucrats who the information that Iraq invaded Kuwait affected.

(128; AE19) The professor did not believe my claim that the report that the school was corrupt was biased.

For example, in (127), only two NPs must be indexed in spec-IP, since *bureaucrats* is removed from the receivers set once it occupies the complement position:

RECEIVERS	spec-IP:	[$_{NP}$ *information*], [$_{NP}$ *Iraq*]

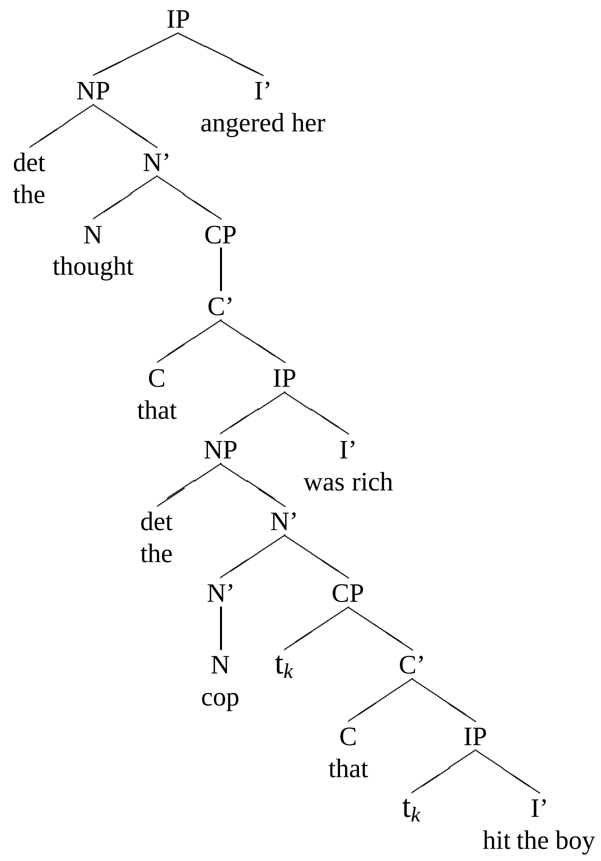


FIGURE 5.9: Embedded complements of nominals.

5.2.5 Clefts

The cleft construction in English is a particular kind of predicate complement construction that serves to focus part of the sentence. There are two types. The *cleft* has *it* as its subject and something like a relative clause at the end (129b):

(129) (a) The man saw a dog.
(b) It was a dog that the man saw.

The *pseudo-cleft* has something like a Wh-clause in subject position:

(130) What the man saw was a dog.

NL-Soar correctly predicts that two embedded relative clauses in the complement NP of a cleft should be acceptable:

(131; AE20) It was a dog that the man that the cop admired saw.

In (131), the complement NP *dog* need not be held simultaneously with the two subjects of the embedded clauses (*man*, *cop*). The same prediction holds for clefts with embedded complements of NPs:

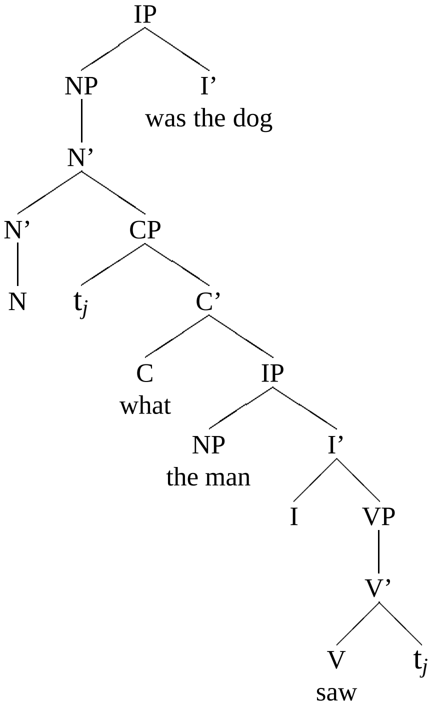


FIGURE 5.10: A pseudo-cleft.

(132; AE21) It was the cop that the information that the dog bit the boy influenced.

Adding one more embedded clause in results in parsing breakdown:

(133; PB12) It is the enemy’s defense strategy that the information that the weapons that the government built didn’t work properly affected.

RECEIVERS	spec-IP:	[$_{NP}$ *information*], [$_{NP}$ *weapons*], [$_{NP}$ *government*]

A clefted subject sentence is unacceptable for grammatical reasons, since the topicalized subject sentence cannot appear in such embedded contexts:

(134; PB13) *It is the enemy’s strategy that for the weapons to work would affect.

The initial Wh-clause in pseudo-clefts is analysed as a headless relative clause. Figure 5.10 shows the structure for (130). The interaction of this structure with the A/R set leads to some interesting predictions. Because the initial Wh-word does not occupy spec-IP position, it should be possible to embed an additional relative clause within the headless relative without causing difficulty:

(135; AE22) What the woman that John married likes is smoked salmon.

NL-Soar handles this because only two NPs need to be simultaneously indexed on one relation:

RECEIVERS	spec-IP:	[$_{NP}$ *woman*], [$_{NP}$ *John*]

A similar prediction holds for an embedded sentential complement:

(136; AE23) What the rumor that the accused man had robbed a bank influenced was the judge's decision.

As predicted, one additional embedded clause does lead to breakdown:

(137; PB14) What the information that the weapons that the government built didn't work properly affected was the enemy's defense strategy.

Sentence subjects in pseudo-clefts are ruled unacceptable for the grammatical reasons discussed above:

(138; PB15) What for the weapons to work properly would affect is the enemy's strategy.

5.2.6 Though-preposing

Though-preposing is another kind of focusing construction which is used to front predicate complements:

(139) Intelligent though the man is, he has no sense of humor.

The phrase structure is given in Figure 5.11. The moved AP occupies spec-CP of the CP headed by *though*. Thus, the fronted constituent does not interfere with buffering NPs on the spec-IP relation, predicting the acceptability of an embedded relative clause (Gibson, 1991):

(140; AE24) Intelligent though the man that Ellen married is . . .

RECEIVERS	spec-IP:	[$_{NP}$ *man*], [$_{NP}$ *Ellen*]

Or an embedded sentential complement:

(141; AE25) Shocking though the news that Iraq invaded Kuwait was . . .

Adding one additional embedded clause leads to breakdown, as predicted:

(142; PB16) Surprising though the information that the weapons that the government built didn't work properly was . . .

Though-preposing with a sentence subject is unacceptable for the grammatical reasons discussed above:

(143; PB17) *Surprising though for the weapons to work properly would be . . .

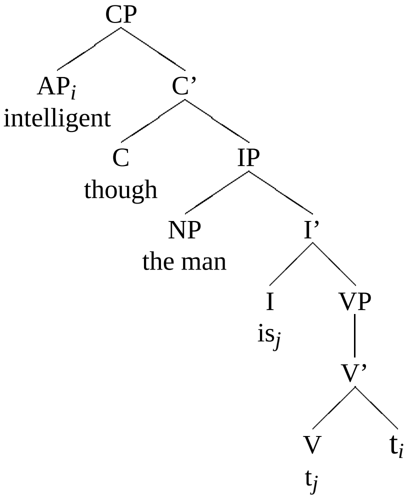


FIGURE 5.11: Though-preposing.

5.2.7 Pied-piping

Pied-piping is a right-branching structure that can be used to avoid stranding prepositions:

(144) (a) *The table which the man put the block which the cop put the dog on on is big.

(b; AE26) The table on which the man put the block on which the cop put the dog is big.

The phrase structure for (144b) is given in Figure 5.12. As Pickering and Barry (1991) point out, pied-piping presents a challenge for parsers that must wait to posit the object traces until their surface positions:

(145) The table on$_i$ which the man put the block on$_j$ which the cop put the dog t$_j$ t$_i$ is big.

If NL-Soar was forced to wait to posit traces, then pied-piping would become unacceptable. The complement relation in the assigners set would eventually drop verbs, preventing the generation of the traces. However, there is no reason NL-Soar must wait for the surface position to posit the trace. As discussed in §3.3.3, the trace is generated as soon as the antecedent and the structural assigner are available. Thus, NL-Soar correctly predicts that, like right-branching, pied-piping may be continued indefinitely:

(146) We saw the the table on which the man put the block on which the cop put the dog on which the boy put the collar . . .

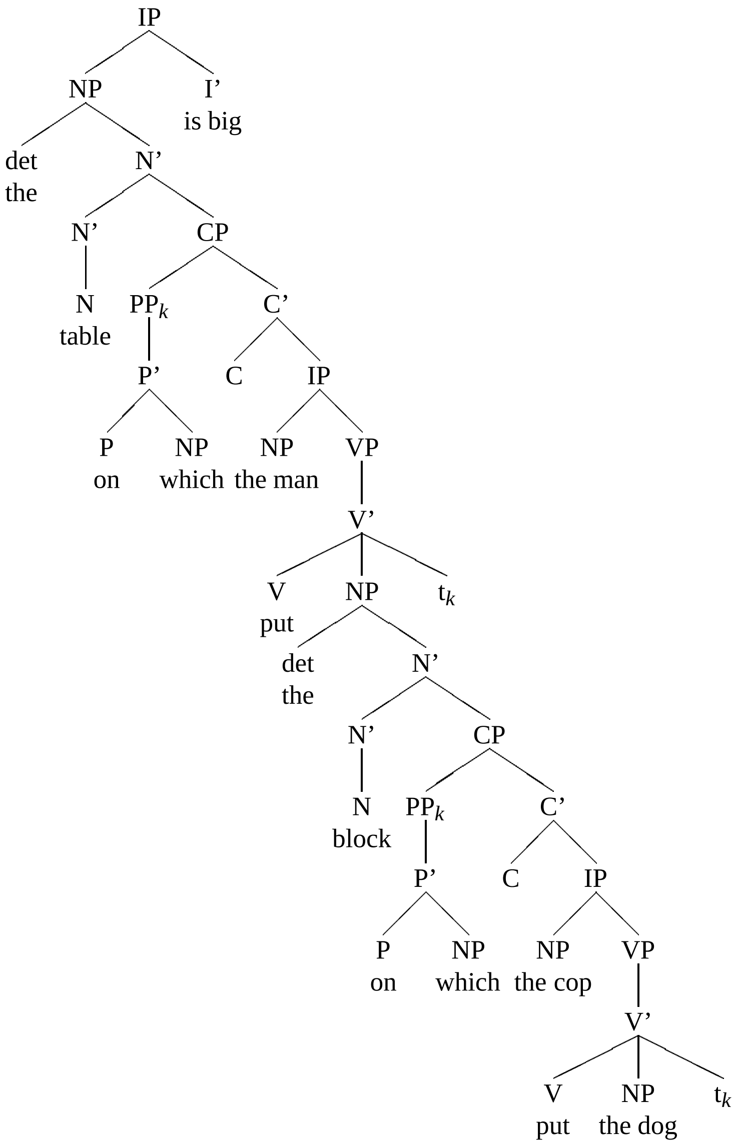


FIGURE 5.12: Pied-piping.

1. Independence of ambiguity
2. Insufficiency of embedding depth alone to cause parsing breakdown
3. Fairly sharp drop in acceptability at two center-embeddings
4. Independence of length
5. Independence of short-term item memory
6. Little effect of explicit instruction and training
7. Some effect of semantic content

FIGURE 5.13: Qualitative phenomena of parsing breakdown (from Chapter 2).

5.3 Accounting for the major qualitative phenomena

Chapter 2 presented a list of seven qualitative phenomena surrounding parsing breakdown, summarized in Figure 5.13. Most of these phenomena are accounted for automatically as a result of NL-Soar's performance on the AE/PB collection; others require some additional plausible assumptions that are consistent with the model.

Independence of ambiguity. Parsing breakdown can happen in NL-Soar independently of ambiguities. The contents of the A/R set are not strictly a function of how ambiguities are resolved, but a function of the interfering properties of the incoming syntactic structure. In the classic difficult center-embedding, all of the subject NPs are not available because the A/R set cannot index them all simultaneously. Ambiguity is not an issue[3].

Insufficiency of embedding depth alone to cause breakdown. The A/R set places no limits on the amount of phrasal embedding. The limit is strictly in the buffering of identically-indexed phrases. This may or may not lead to limits on embedding, depending on the syntactic structure. In particular, there is no limit on right- or left-branching, while there are severe limits on center-embedding.

Fairly sharp drop in acceptability of center-embeddings. The empirical evidence presented in Chapter 2 made clear from a range of measures that doubly center-embedded object-relatives were unacceptable, while singly-embedded object-relatives are acceptable. NL-Soar predicts this sharp drop in acceptability simply because it is the third NP in the doubly-embedded construction that exceeds the A/R set capacity.

Independence of length. The limitations of the A/R set are a function of the syntactic structure of the input, not its length. We saw above that short sentences (e.g., PB2) can lead to breakdown, while long sentences are acceptable (e.g., AE26). In general, NL-Soar places *no* limit on the length of acceptable sentences.

Independence of short-term item memory. Parsing breakdown can occur even though subjects are able to recall all the items in the sentence (Larkin & Burns, 1977). The short-

[3] Of course, we saw in Chapter 4 that the A/R set does place limitations on the ability to detect and resolve ambiguities, as well.

term memory accessed in this case is very likely to be a phonological memory (Baddeley, 1990). The limitations imposed on the A/R set have nothing to do with any limitations of the phonological buffer. Thus, it is possible to find short sentences which overload the syntactic processing without overloading the phonological buffer. This is precisely what happens in the case of short center-embeddings: the items can be recalled, but cannot be parsed.

Little effect of explicit instruction and training. The structure of the A/R set is an architectural hypothesis; it cannot be modulated by knowledge. Thus, NL-Soar predicts that the limitations that arise from the structure of the A/R set cannot be overcome without using some auxiliary memories. One such possible auxiliary memory is paper and pencil—eventually very deep embeddings can be worked out with enough time and patience. But there is another more interesting possibility: the phonological buffer. If, as argued above, the phonological buffer can maintain a short double center-embedding, then with sufficient practice it may be possible to deliberately work out the correct pairings of the words based on this memory alone. In fact, the subjects of Blauberg & Braine (1974) were able to increase their ability to comprehend center-embeddings by one level—that it, they eventually learned to comprehend double embeddings, but not triple embeddings. The plausible explanation of this result is that the triple embeddings exceeded the short-term capacity of the phonological buffer, so the newly learned pairing skill could not be applied. Note that working out the correct pairing is all that is required to succeed on the comprehension test; it is not at all clear that the subjects also learned to perceive the structure as grammatical (Marks, 1968).

Effect of semantic content. Comprehension of semantically supported center-embeddings is better than semantically neutral center-embeddings (Stolz, 1967). NL-Soar cannot account for this result within the confines of the A/R set for the reasons mentioned above. However, all that is needed to succeed in these comprehension tests is some memory of the items in the sentence, combined with general knowledge that permits plausible relations to be established between the items. For short sentences of the kind used in the studies, the phonological short-term memory may suffice, or perhaps the partial situation model constructed from the comprehension of the nouns and verbs. In either case, semantic knowledge could be brought to bear to produce a plausible pairing of items without the A/R set playing a significant role.

5.4 Summary and general discussion

Table 5.1 summarizes the predictions on the collection of parsing breakdown and acceptable structures. The results are good: the theory accounts for 39 of 43 structures (91%) (with the ungrammatical cases removed from consideration, the results are 33 of 36, or 89%) and as pointed out above, some of the missed predictions involve somewhat marginal judgments. Only the theory of Gibson (1991) has comparable coverage, and no existing *architectural* theory approaches such detailed coverage. Furthermore, all seven major qualitative phenomena can be accounted for with the additional assumption of a phonological buffer, for which there is independent evidence.

TABLE 5.1: Summary of predictions on AE/PB collection.

Right and left branching	AE1	●		
	AE2	●		
Center-embedded relatives	AE3	●	PB1	●
	AE4	●	PB2	●
	AE5	●	PB3	●
	AE6	●	PB4	○
	AE7	●		
Subject sentences and topicalizations	AE8	●		
	AE9	●	PB5	●
	AE10	●	PB6	●
	AE11	●	PB7	●
	AE12	●	PB8	○
	AE13	●		
Complements of nominals	AE14	●		
	AE15	○	PB9	●
	AE16	●	PB10	●
	AE17	●	PB11	○
	AE18	●		
	AE19	●		
Clefts	AE20	●	PB12	●
	AE21	●	PB13	●
	AE22	●	PB14	●
	AE23	●	PB15	●
Though-preposing	AE24	●	PB16	●
	AE25	●	PB17	●
Pied-piping	AE26	●		

●= correct prediction
○= incorrect prediction

The predictions derive from an interaction of the two-valued indices of the A/R set with the syntactic structures assigned by Government and Binding theory. A general discussion of the A/R set and the magic number two must wait until Chapter 9, but it is worth commenting here on the role of syntactic theory in the model. The model can certainly be formulated in a way that abstracts away from the particular choice of syntactic relations. For example, an A/R set model can easily be constructed that uses traditional grammatical functions such as subject and object, rather than structural positions such as spec-IP. The advantage to using an existing syntactic theory, whatever its form, is that the set of relations are not posited to simply account for the performance data, but are motivated by their ability to account for linguistic regularities across languages. Alternative syntactic theories might often lead to the same predictions (e.g, on breakdown on center-embedded relatives), but this need not be the case. For example, the dependency grammar of Mel'ĉuk

(1988), with its ontology of 41 surface grammatical relations for English, may very well increase discriminability enough to change some predictions of breakdown to predictions of acceptability.

While it would be worthwhile exploring alternative syntactic theories, such a task is a substantial undertaking. Changing the underlying syntax will have potential ramifications in the predictions for garden path effects (Chapter 6) and ambiguity resolution (Chapter 4) as well as parsing breakdown. At this point, the most that can be concluded is that the Government and Binding structures do a good job of helping make the right predictions, and it is not clear that any alternative syntactic theory would do significantly better. In fact, we have seen several cases (such as subject sentences) where the precise analyses of GB lead to rather interesting predictions that would not necessarily be captured by other approaches.

Perhaps the most significant open issue for the theory is establishing the nature of the strategies for resolving conflicts in the A/R set. The issue was effectively avoided here by using the theory in a way that abstracts away from the effects of particular strategies. But the questions remain: Which strategies are the right ones? Are the strategies learned, or architectural? The question is an interesting one: we saw in Chapter 4 that the obvious alternative for an architectural strategy (pure recency) is unlikely to be correct.

Beyond the few missed predictions and the origin of the A/R set strategies, the biggest issue for further development of the theory is empirical data. The collection used here is primarily based on informal linguistic acceptability judgments, and some of the judgments involve somewhat questionable borderline cases. Unfortunately, the psycholinguistic evidence concerning parsing breakdown is almost exclusively concerned with center-embedded object relatives. Before attempting to modify the theory to increase the empirical coverage, it seems worthwhile to develop a better empirical database.

In the next chapter, we consider parsing difficulty that is caused by local ambiguity, rather than deep embedding.

Chapter 6

Garden Path Effects and Unproblematic Ambiguities

ALTHOUGH GARDEN PATH EFFECTS have played a significant role in psycholinguistics over the past two decades, the appearance of detailed theories of a wide range of both garden paths (GP) and unproblematic ambiguities (UPA) is relatively recent, starting with Pritchett's 1987 thesis. This chapter presents NL-Soar's account of GP and UPA phenomena. The first section describes the NL-Soar garden path theory, derived directly from the structure of the model presented in Chapter 3. Then the theory is applied in detail to the 57-item collection of garden paths and unproblematic ambiguities. The third section shows how NL-Soar accounts for the major qualitative garden path phenomena. The chapter concludes with a discussion and summary of the results.

6.1 The NL-Soar theory of garden path effects

The NL-Soar garden path theory can be summarized as follows. Comprehension is essentially a single path process, with a limited capability to *recognitionally repair* inconsistencies that may arise in the syntactic structure when the wrong path is taken. When recognitional repair fails, a garden path effect occurs.

NL-Soar's repair mechanism is *simple destructive repair* (§3.3.3). The mechanism consists of the primitive utterance model constructors plus the snip operator. A repair happens when snip breaks an existing relation in the utterance model, and the utterance model is reconstructed with the standard link operators. Through Soar's chunking, this repair process becomes a seamless part of the recognitional repertoire of comprehension operators (§3.3.4).

Snip responds to inconsistencies detected in the utterance model. The inconsistencies that can arise are:

1. Attachments to competing syntactic senses of the same lexical token.

2. Attachments of multiple nodes to the same structural position.

3. Missing obligatory structure.

In case 1, snip operators are generated to remove relations to one of the competing lexical senses. In cases 2 and 3, snip operators are generated to remove relations local to the detected inconsistency, where local is defined to mean within the containing maximal projection. We will see examples of all these cases shortly.

This kind of repair can also be characterized as *cue-based*: certain structural cues trigger the repair process. This tightly constrained generation of potential snip sits ensures computational efficiency in both problem space search and knowledge search (match in the recognition memory).

What does it mean for NL-Soar to experience a garden path? Any garden path theory must ultimately be a theory of conscious processing difficulty that manifests itself as an impression of ungrammaticality. However, the Soar architecture does not make commitments about what is in conscious awareness (Newell, 1990). It is therefore necessary to make an additional assumption about what gives rise to conscious processing difficulty in order to make predictions about garden path sentences:

(147) *The NL-Soar garden path assumption:* A garden path effect arises in NL-Soar when it cannot recognitionally repair an inconsistent utterance model.

This an obvious and straightforward assumption, but it is important to be explicit about it, since without it no garden path predictions can be made.

Given this assumption, there are three possible ways that garden path effects might emerge in NL-Soar:

1. The appropriate structural cues are not available to trigger the repair.

2. The syntactic relations that must be altered (snipped) are no longer available in the A/R set.

3. The particular repair sequence that is required has not yet been chunked as part of a top level comprehension operator. As a result, an impasse will arise and more deliberate behavior will be required to repair the utterance model.

In all three cases, the *severity* of the garden path effect depends on how much deliberate processing is required for recovery. This means that, like any other chunked skill in Soar, the efficacy of syntactic recovery (both on-line repair and deliberate reprocessing) is partly a function of experience. Although this is an interesting prediction in itself, the application of the theory in the remainder of this chapter will abstract away from the effects of experience by assuming that the repair processes *are* chunked, leaving the burden of making garden path predictions on cases 1 and 2. For adult comprehenders, this is a reasonable assumption to make in most cases (though it does raise the possibility that the theory may overpredict acceptability). Furthermore, we shall not be concerned here with accounting for differences in garden path severity, though such differences do exist (e.g., (Warner & Glass, 1987)).

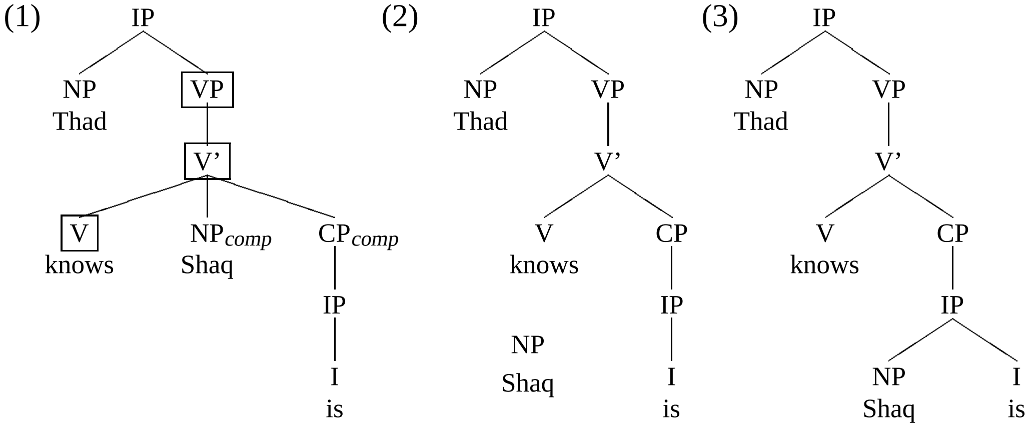


FIGURE 6.1: Repairing an unproblematic subject/object ambiguity.

Any impression of ungrammaticality that arises, no matter how rapidly it is recovered from, is interpreted as a failure of recognitional comprehension, and thus a garden path effect.

As an example of how the theory predicts a garden path, consider the contrast between the following object/subject ambiguities: (148) is perfectly acceptable, while (149) causes a garden path effect (Frazier & Rayner, 1982). Recall that all unproblematic ambiguities are represented by a pair of sentences, neither of which causes any difficulty[1].

(148; UPA1) (a) Thad knows Shaq.
(b) Thad knows Shaq is tall.

(149; GP1) Since Jay always jogs a mile seems like a short distance to him.

Figure 6.1 reviews the repair process for (148), which was first described in §3.3.3. *Shaq* is initially attached in complement position of *knows*. When *is* arrives, it is projected to a CP and attached in complement position, since *knows* can take a sentential complement. Now there is an inconsistency in the phrase structure: two nodes occupying the same structural position. A snip is generated to break a structural relation local to the inconsistency: the complement relation between [$_{V'}$ *knows*] and [$_{NP}$ *Shaq*]. Next, [$_{NP}$ *Shaq*] is attached in its proper final location as the subject of [$_{IP}$ *is*]. The boxed nodes in the Figure 6.1 identify the maximal projection in which the inconsistency is detected. In other words, these nodes delimit the scope of consideration for generating snip operators. Only relations that involve one of these nodes are considered for snipping.

In (149), *a mile* is taken initially as the complement of *jogs*, just as in (148). 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

[1] See (Gibson, 1991) and (Frazier & Rayner, 1982) for a discussion of why sentences like (149) are grammatical without the comma.

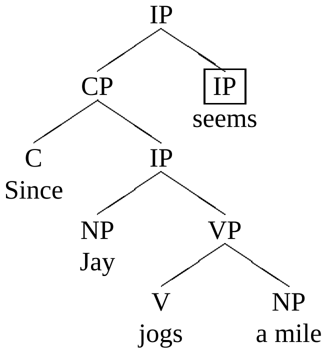


FIGURE 6.2: Failure to repair a subject/object ambiguity.

inconsistency (the missing obligatory subject). This situation is shown in Figure 6.2, with the boxed nodes again representing the locality of the inconsistency. As a result, [$_{NP}$ *a mile*] cannot be reanalysed as the subject of [$_{IP}$ *seems*], and the necessary repair fails[2].

This garden path theory has a number of distinguishing features. First, 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. This chapter is concerned purely with the repair process, which determines, given a particular structural interpretation that is inconsistent with incoming input, whether or not that structure can be repaired. Whether a local ambiguity gives rise to a garden path effect in any particular context is a function both of the ambiguity resolution process itself, and the efficacy of the repair. This chapter is concerned only with the latter process; Chapter 4 is devoted to NL-Soar's theory of ambiguity resolution.

Second, the theory is a *functional* theory, in that it posits mechanisms to efficiently carry out the functions of comprehension. NL-Soar is not a metric that distinguishes garden paths from non-garden paths. It classifies certain sentences as (potential) garden paths because it may fail to recognitionally parse those sentences.

Third, the theory embodies the *Structural Garden Path Hypothesis* (54), which states that GP effects are a function of differences between the syntactic structure of the preferred interpretation, and the syntactic structure of the correct interpretation. NL-Soar embodies this hypothesis because GP effects arise from limitations in a repair process that maps the syntactic structure of one interpretation into the syntactic structure of another interpretation.

Finally, the model works *without reprocessing* the input. Recognitional repair happens rapidly and rather frequently, without the need to reread or rehear the input.

[2] Warner and Glass (1987) actually discovered an instance of a short version of (149) that most of their subjects found acceptable, as measured by a rapid grammaticality judgment task:

(150; UPA2) (a) When the boys strike the dog kills.
(b) When the boys strike the dog the cat runs away.

NL-Soar still predicts this to be a GP. One possible explanation of this result is that the intransitive use of *kills* somehow facilitates the recovery of the intransitive use of *strike*, which is required for the correct analysis. It is not clear where the locus of such an effect should be in NL-Soar. For now, it is simply a missed prediction.

6.2 Predictions on the GP/UPA collection

To reiterate a point made above, the predictions on the GP/UPA collection will be made independently of the presumed preferred direction of resolution of the ambiguity. Despite this independence, the theory is still constrained by the data in the following way. For each GP sentence, there must exist a grammatical partial path in the sentence such that the system cannot repair from that path to the correct interpretation. For each UPA pair, there must exist a single grammatical partial path such that the system can obtain a correct interpretation in both cases, because it chooses the correct path for one case and chooses the wrong path but can recover in the other case.

The predictions below are grouped by ambiguity type, with both GP and UPA items considered in each group. Many of the predictions are illustrated with annotated phrase structure trees. A summary of the results appears at the end of the chapter.

6.2.1 Object/subject and object/specifier ambiguities

We have already seen how NL-Soar handles two cases involving object/subject ambiguities ((148) and (149) above). This section explores a range of cases involving direct and indirect objects, prepositional objects, NP specifiers, and clausal subjects.

The distance between the ambiguous point and the disambiguating material in structures like (148) can be rather extended without causing difficulty (Pritchett, 1992):

(151; UPA3) (a) Ron believed the ugly little linguistics professor.
(b) Ron believed the ugly little linguistics professor he had met the week before in Prague disliked him.

As long as the complement relation assigned by the relevant verb (in this case *believe*) is still available in the A/R set, then NL-Soar can repair the structure regardless of the length of the object nounphrase. The repair mechanism is only sensitive to the syntactic structure, not the surface string.

However, Warner and Glass (1987) *did* manage to produce an apparently length-induced garden path effect, using exactly the same object/subject ambiguity:

(152; GP2) The girls believe the man who believes the very strong ugly boys struck the dog killed the cats.

Surprisingly, NL-Soar accounts for such garden path effects. The intervening material is such that all the structural relations required for a successful repair cannot be held in the A/R set. In (152), *The man* and *the boys* are taken as complements of *believe* and *believes*, respectively. When *struck* arrives, it must be placed on the comp-V' assigners relation so that it may take its complement. *Believes* must also still be available on the comp-V' relation, because the complement of *believes* must change from *the boys* to *struck*. However, *believe* must also be on the comp-V' relation, because the complement of *believe* must be changed from *the man* to *the man killed.* For the repairs to be successful, the A/R set needs to support the following structure:

ASSIGNERS	comp-V':	[$_{V'}$ *believes*], [$_{V'}$ *believe*],[$_{V'}$ *struck*]

This exceeds the posited capacity of two nodes per relation (Chapter 5), which means that at least one of the complement relations will not be available. Thus, the interaction of the repair mechanism with the limited syntactic working memory produces a garden path effect because the relevant structural relations are not available to snip. The important factor is not the *length* of the intervening material, but the interaction of the *structure* of the intervening material with the structure to be repaired.

Another kind of difficulty object/subject ambiguity arises with embedded subject sentences (Gibson, 1991):

(153; GP3) Aarti believes that John smokes annoys me.

[$_{CP}$ *That John smokes*] is taken as the complement of *believes* but must be reanalysed as the topicalized subject of [$_{IP}$ *annoys*] (see §5.2.3 for a discussion of the syntax of subject sentences). A similar A/R set overload explanation can be given for this garden path. *Believes*, *smokes* and *annoys* are all placed on the comp-V' assigners relation, and assuming that *annoys* displaces the less recent *believes*, the crucial complement relation from [$_{V'}$ *believes*] to [$_{CP}$ *that John smokes*] cannot be repaired. Unlike the explanation for (152), this requires an assumption about the strategy for managing the comp-V' relation (an assumption that, nonetheless, is implemented in the system and is consistent with the other results presented here). However, it is not clear that a processing explanation should even be sought for the unacceptability of (153), because the *unambiguous* version of the construction is also unacceptable:

(154) ?Aarti believes that that John smokes annoys me.

Thus, while an account is possible with NL-Soar, the dubious grammatical status of these embedded tensed subject sentences makes the point somewhat moot.

Fronted clauses with embedded relatives can also produce garden path effects (Warner & Glass, 1987):

(155; GP4) Before the boy kills the man the dog bites strikes.

Even if the clause *the dog bites* is properly interpreted as a reduced relative modifying *the man*, when *strikes* arrives and is adjoined to the fronted clause (as in Figure 6.2), the relevant snip operator is not generated to detach *the man* from *kills*, for the reasons given above in example (149). Because the misinterpretation of the reduced relative is not necessarily the critical factor, the theory correctly predicts that the garden path will persist even if the relative clause is disambiguated:

(156) Before the boy kills the man that the dog bites strikes.

Warner and Glass produced an interesting twist on this garden path. They managed to contextually induce the *subject* reading of the ambiguous NP. As a result, an otherwise acceptable construction became a garden path:

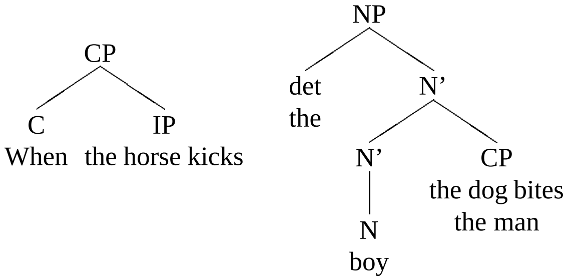


FIGURE 6.3: A reverse subject/object garden path.

(157; GP5) When the horse kicks the boy the dog bites the man.

Because the theory can be applied independently of the initial choice, we can determine if NL-Soar predicts a garden path effect when *the boy* is not taken as the object of *kicks*. In fact, it does. *The boy the dog bites* is taken as a complex NP with a relative clause modifier. *The man* is attached as the complement of *bites*, displacing the posited object trace (via a snip—we will see other examples of repairs involving traces in §6.2.5). At this point, the end of the sentence is reached but no snip operators are generated since there are no local inconsistencies (this is true even if *the boy* is finally attached as the complement of *kicks*). This situation is shown in Figure 6.3.

Objects can be reanalysed to NP specifier position without difficulty:

(158; UPA4) (a) Without her we failed.
(b) Without her contributions we failed.

(159; UPA5) (a) The cop saw her.
(b) The cop saw her sons employees[3].

The structure in (159) involves reanalysis of an ambiguous plural/genitive NP from object to specifier position; (158) involves a pronoun that can take objective or genitive case. Figure 6.4 shows the detected inconsistency that arises and the subsequent repaired structure for (158).

A similar ambiguity does cause a garden path effect (Frazier, 1978; Pritchett, 1988):

(160; GP6) Without her contributions failed to come in.

Her is initially taken as the object of *without*, then reanalysed as the specifier of $[_{NP}$ *her contributions*] as in (158) above. Then the PP $[_{PP}$ *without her contributions*] is adjoined to $[_{CP}$ *failed*]. To perform the correct repair, snip operators must be generated to remove $[_{NP}$ *her*] from specifier position, and remove $[_{NP}$ *contributions*] from object position. As in example (149), these snips are not generated since they are not local to the detected inconsistency (the missing subject of $[_{IP}$ *failed*]).

Garden path effects can also be created with double-complement verbs (Pritchett, 1992):

[3] This sentence is incorrect as printed text without the apostrophe. The omission of the apostrophe simply preserves the ambiguity found in speech.

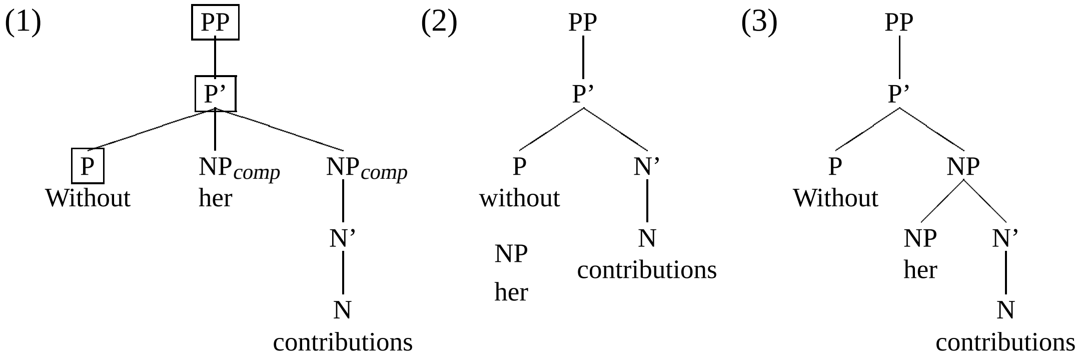


FIGURE 6.4: Repairing an unproblematic object/specifier ambiguity.

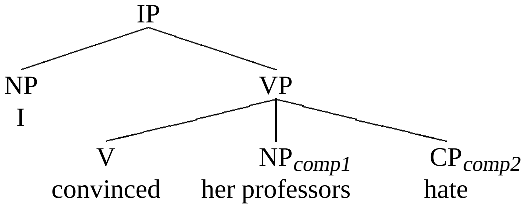


FIGURE 6.5: A garden path involving a double complement.

(161; GP7) I convinced her professors hate me.

(162; GP8) The doctor warned the patient would be contagious.

Convinced takes two complements: a nominal first object and a sentential second object (*I convinced her that professors hate me*). In (161), [$_{NP}$ *her professors*] is taken as the first complement of *convinced*. When *hate* arrives, it is projected to CP and attached as the second complement, as shown in Figure 6.5. Because the two structures are *not* occupying the same structural position, no snip operator is generated to detach [$_{NP}$ *her professors*], and the repair fails. A similar explanation holds for (162). The difference is that the nominal complement of *warned* is optional (*The doctor warned he would not tolerate the disruption*). [$_{NP}$ *The patient*] is initially attached in first complement position, and [$_{CP}$ *would*] attached in second complement position. The repair thus fails as in (161).

Another kind of garden path arises when the first object of a double object construction is modified by a relative clause:

(163; GP9) John gave the boy the dog bit a dollar.

In (163), *the dog* is initially attached as the second object of *gave*. *Bit* is projected to CP, but cannot attach to the existing structure. No snip operators are generated since there are no local inconsistencies. When *a dollar* arrives, it is attached as the complement of *bit*, but that only exacerbates the garden path effect. Figure 6.6 shows the final result.

Not all ambiguities involving double objects cause difficulty:

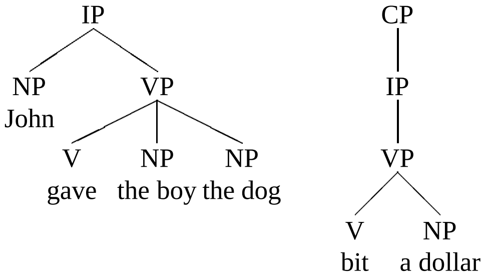


FIGURE 6.6: A double-object/relative garden path.

(164; UPA6) (a) The cop gave her earrings.
(b) The cop gave her earrings to the dog.

Figure 6.7 traces what happens. In (164b), *her* and *earrings* are initially attached in first and second complement positions, respectively. When *to* arrives, it is attached in second complement position. This triggers a snip to remove *earrings*. Next, *earrings* is attached in first complement position, triggering a snip to remove *her*. Finally, *her* is attached as specifier of *earrings*, and the repair is complete. This is the first example of a repair involving more than one snip.

Object/object ambiguities arise when a nounphrase may be interpreted as the object of an embedded clause or the second object of the main clause:

(165; GP10) Anurag gave the man who was reading the book.

Sentence (165) does give rise to a garden path effect if the ambiguous NP (*the book*) is incorrectly taken as the complement of the lower clause. As Pritchett (1992) points out, preferences for how the ambiguity is resolved vary considerably, so the construction does not produce garden path effects as reliably as many of the other constructions discussed in this section. However, that is irrelevant to applying the NL-Soar theory. The question is: *If* the final NP is taken as the object of the lower clause, does a garden path effect emerge? In fact, it does; Figure 6.8 shows the result. The explanation of the effect is simply that there are no local inconsistencies to trigger the snip operator, so no repair takes place.

Not all object/object ambiguities lead to garden path effects. Unproblematic object/object ambiguities may be constructed with complements of nominals (Gibson, 1991):

(166; UPA7) (a) The minister warned the president of the danger.
(b) The minister warned the president of the republic of the danger.

In (166b), [$_{PP}$ *of the republic*] may be initially attached as a complement of [$_{N'}$ *warned*]. When [$_{PP}$ *of the danger*] arrives and is attached in the same complement position, it triggers a snip operator to remove [$_{PP}$ *of the president*], which is then attached as a complement of [$_{N'}$ *president*.]

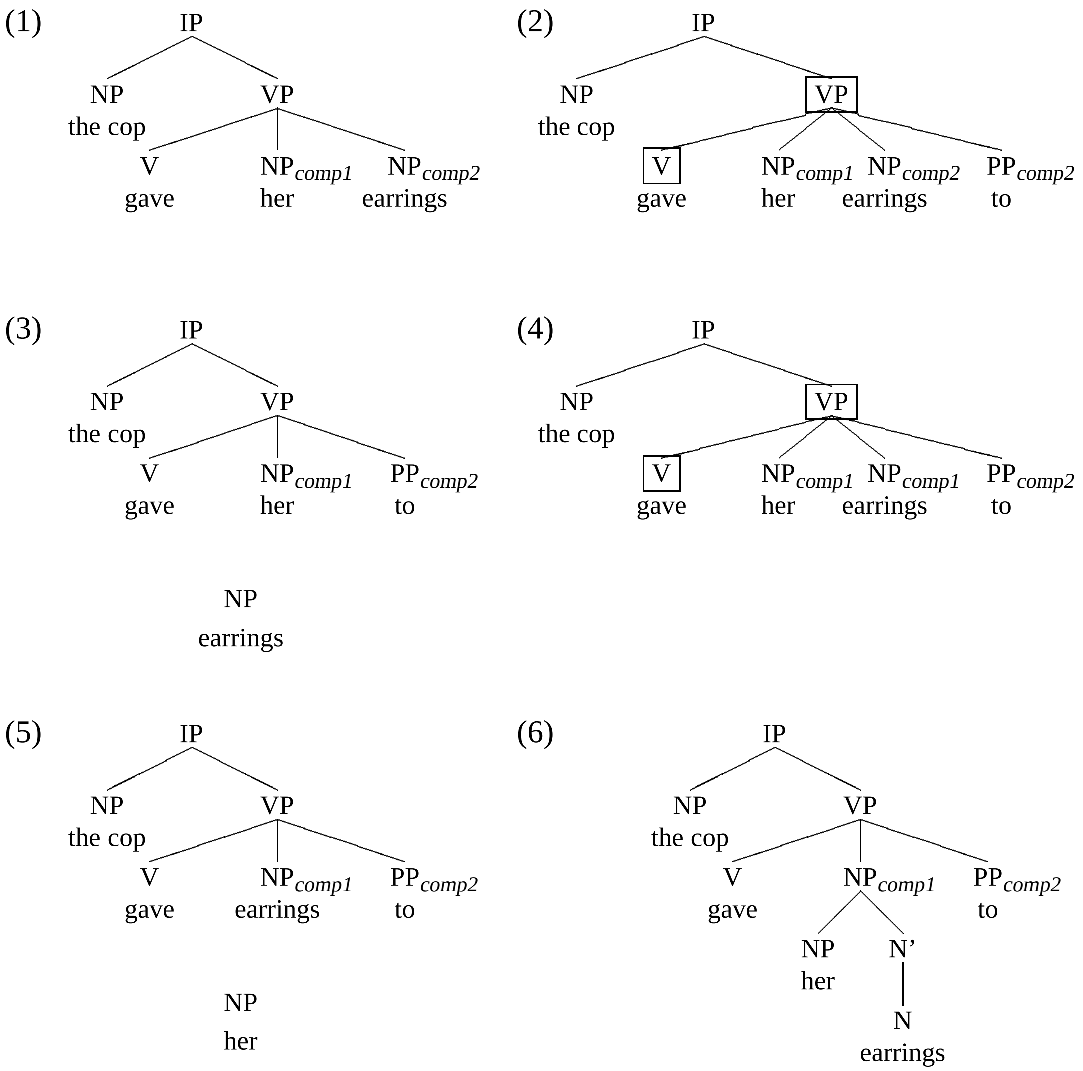


FIGURE 6.7: Repairing a double object ambiguity.

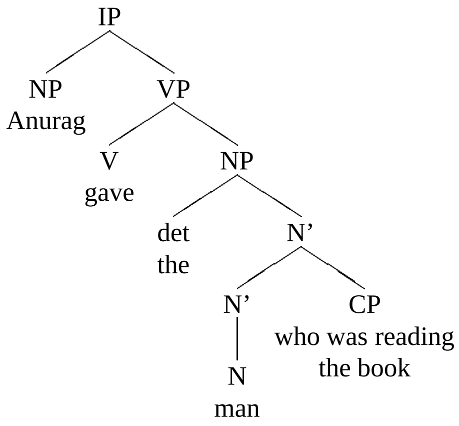


FIGURE 6.8: An object/object garden path.

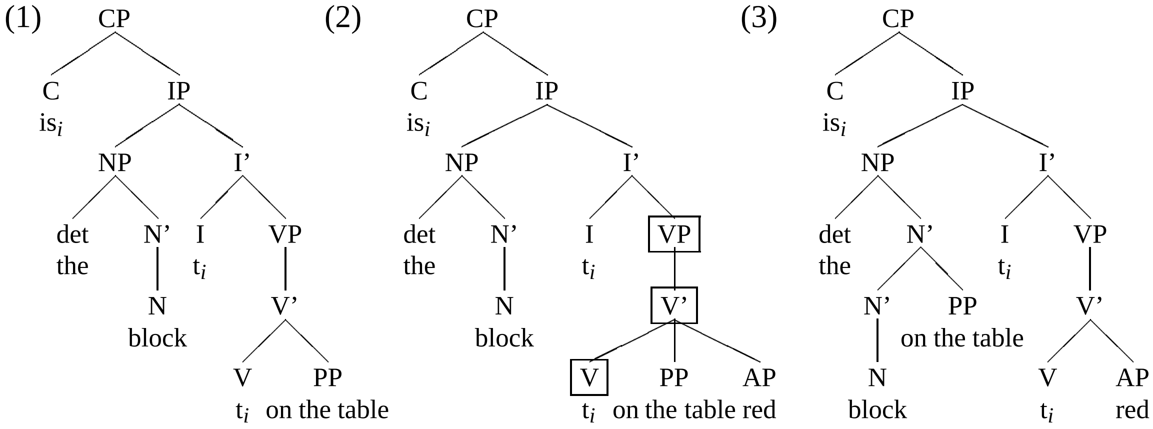


FIGURE 6.9: Repairing an unproblematic complement/adjunct ambiguity.

6.2.2 Complement/adjunct ambiguities

Incoming phrases can often be interpreted as either complements or adjuncts. Consider the unproblematic sentences in (167):

(167; UPA8) (a) Is the block on the table?
(b) Is the block on the table red?

The prepositional phrase *on the table* may be interpreted as a modifier of *block* or the complement of *is*. Assume that the complement attachment is pursued first. Figure 6.9 shows how NL-Soar repairs the structure when *red* arrives, so that the PP [$_{PP}$ *on the table*] is reanalysed as a modifier of *block*. *Red* is projected to an AP and attached in complement position, which triggers a snip operator to detach [$_{PP}$ *in the box*], in the same manner that we have seen above. Once detached, [$_{PP}$ *on the table*] is simply adjoined to [$_{N'}$ *block*]. A similar explanation predicts the acceptability of complement clause/subject-relative ambiguities[4] (Gibson, 1991):

(168; UPA9) (a) John told the man that Mary kissed Bill.
(b) John told the man that kissed Mary that Bill saw Phil.

However, other complement/adjunct ambiguities do cause difficulty. Crain and Steedman (1985) found that complement clause/object-relative ambiguities produce garden path effects:

(169; GP11) The psychologist told the wife that he was having trouble with to leave.

Prepositional phrase argument/adjunct ambiguities may also produce garden path effects (in contrast to (167) above) (Gibson, 1991):

[4] The repair could be avoided entirely by following Pritchett's (1992) assumption that CPs are not licensed until their constituent IPs are present.

(170; GP12) I sent the letters to Ron to Teresa.

Unfortunately NL-Soar cannot account for these garden paths, because the repair succeeds in both cases as in (167) above. However, the complement/adjunct ambiguities exemplified by (170) seem to differ in their acceptability:

(171) ?Michael put the toys in the bag into the closet.

Such examples demonstrate the complexity of PP attachment phenomena.

Crain and Steedman (1985) also found that a garden path arises when the clause is interpreted as a relative clause but the complement reading is required (they induced the relative reading through a contextual manipulation):

(172; GP13) The psychologist told the wife that he was having trouble with her husband.

NL-Soar does account for this garden path. Even if [$_{NP}$ *her husband*] is attached as the complement of [$_{P'}$ *with*], the critical snip operator required to detach the clause from [$_{N'}$ *wife*] is not generated.

Gibson (1991) points out the following unproblematic ambiguity involving complements of nominals:

(173; UPA10) (a) The report that the president sent to us helped us make the decision.
(b) The report that the president sent the troops into combat depressed me.

The clause *that the president sent...* may be taken as the complement or modifier of *report*, and neither interpretation causes difficulty. NL-Soar fails to predict this. If the clause is attached as the complement, the relevant snip (to break the complement link) is not generated when the missing object of *sent* is detected, because the relation is not local to VP [$_{VP}$ *sent*]. Similarly, nothing triggers the snip to break the adjunction relation if the clause is first attached as a modifier.

Another kind of unproblematic complement/adjunct ambiguity involves adjectives that may be taken as predicate complements or modifiers of incoming nouns:

(174; UPA11) (a) The boy got fat.
(b) The boy got fat mice for his pet snake.

Figure 6.10 shows how NL-Soar repairs such constructions. [$_{NP}$ *Mice*] is attached as the complement of [$_{V'}$ *got*], triggering the snip of [$_{AP}$ *fat*], which is then adjoined to [$_{N'}$ *mice*].

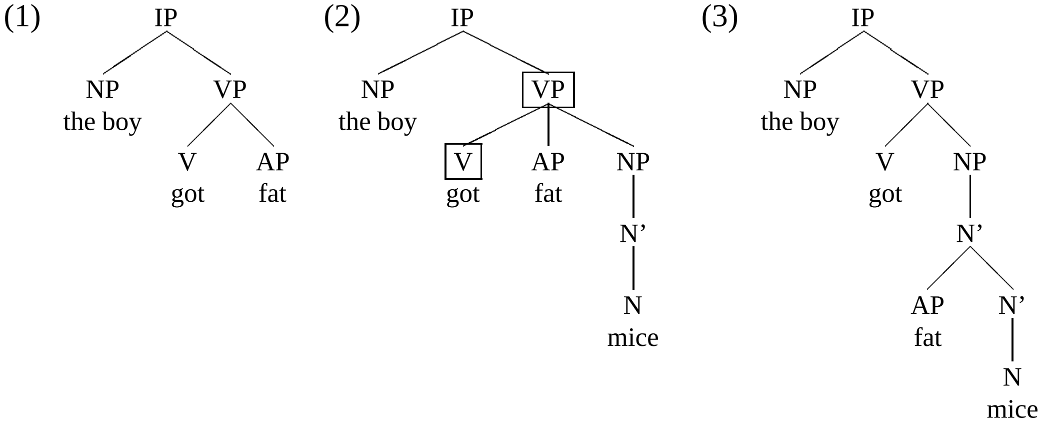


FIGURE 6.10: Repairing an unproblematic predicate complement/modifier ambiguity.

6.2.3 Main verb/reduced relative ambiguities

We now explore variations on the reduced relative garden path. Consider the canonical example:

(175; GP14) The horse raced past the barn fell.

Figure 6.11 the shows complete structure for the main verb interpretation of *The horse raced past the barn.* The inflectional features that head the IP phrase adjoin to the zero-level verb node, leaving a trace in head of IP. (This joining of inflectional features to the verb is assumed in some form by most syntactic theories; e.g., McCawley (1988) calls it *Tense-hopping*, and assumes an adjunction structure like the one presented here.)

Passive forms like *driven* are untensed. Figure 6.12 shows the reduced relative reading of *The horse raced past the barn*, which uses the passive interpretation of *raced*. In this structure, the inflection plays no role.

Consider the repair required to successfully parse (175). The structure in Figure 6.11 must be transformed into the structure in Figure 6.12. This involves removing [$_{NP}$ *the horse*] from spec-IP position, and 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 6.13. However, this fails to generate all the required snips. Although the spec-IP relation is local to the IP, the crucial inflection adjunction is not, so the passive reading cannot be recovered.

The intervening modifier [$_{PP}$ *past the barn*] is irrelevant to this explanation. Thus, NL-Soar correctly predicts the existence of very short reduced relative garden paths (Kurtzman, 1985; Abney, 1989)[5]:

[5] One possibility here is to allow [$_{V'}$ *sank*] to attach as the head of the existing VP ([$_{VP}$ *floated*]), which would then make the VP, rather than the IP, the locality of the inconsistency. But allowing such links extends the range of existing constructive processes. The basic assumption of simple destructive repair is that the

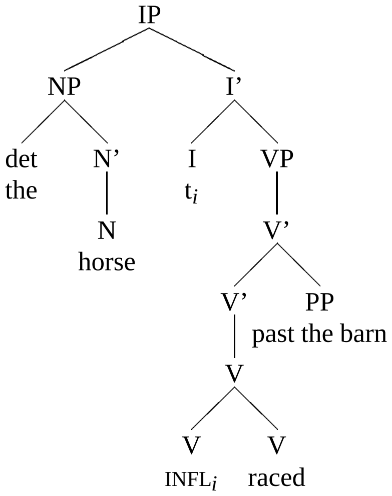


FIGURE 6.11: The main verb reading of *The horse raced past the barn.*

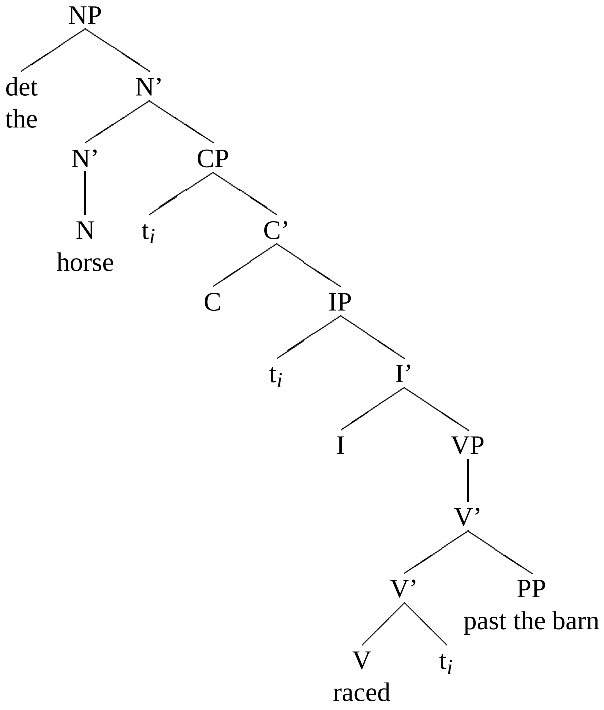


FIGURE 6.12: The reduced relative reading of *The horse raced past the barn.*

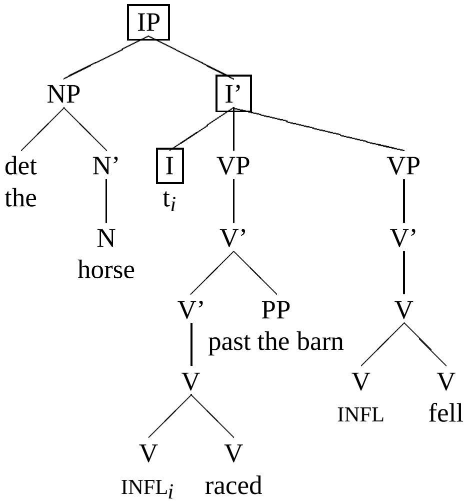


FIGURE 6.13: The main verb/reduced relative garden path.

(176; GP15) The boat floated sank.

The explanation also extends to ditransitive verbs (Rayner et al., 1983):

(177; GP16) My friend sent the flowers smiled broadly.

Embedding the relative clause gives rise to a similar kind of difficult ambiguity (Gibson, 1991):

(178; GP17) The dog that was fed next to the cat walked to the park chewed the bone.

In (178), *walked* must modify *cat*, but unlike (175), a different NP is in subject position (*dog*). Again, however, when *chewed* is projected to VP and attached to the main clause IP, the required snip operators are not generated.

Not all main verb/reduced relative ambiguities produce garden path effects (Pritchett, 1992; Gibson, 1991):

(179; UPA12) (a) The defendant examined the evidence.
(b) The defendant examined by the lawyer shocked the jury.

The reduced relative reading in (179b) is readily available, in stark contrast to (175). NL-Soar handles the repair in the following way. The incoming preposition *by* projects to PP and adjoins to *examined*, just as *past* adjoins to *raced* in (175). The crucial difference is that *examined* is obligatorily transitive. This means that $[_{V'}$ *examined*$]$ is missing an object, which must normally immediately follow the verb[6]. Figure 6.14 shows a snapshot

existing set of constructive processes (plus snip) are sufficient. Structures of the form $[_{VP}\ [_{V'}\ [_{V}\ ]]]$ are always produced by projection. Thus, the incoming verb will create its own projection rather than attaching as the head of a different verb's projection.

[6] This adjacency requirement (sometimes attributed to Case assignment) can easily be seen in the awkwardness of examples such as:

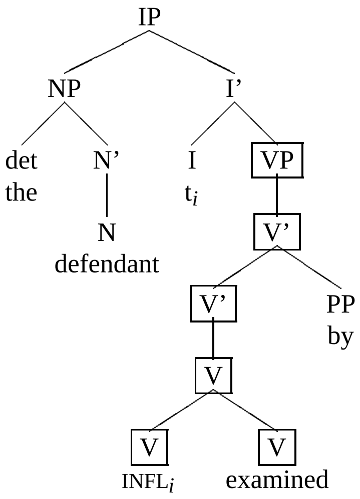


FIGURE 6.14: An unproblematic reduced relative ambiguity.

at this point. Two snips are generated local to the VP: a snip that removes the adjoined inflection, and a snip that detaches the VP from the I'. Now [$_{VP}$ *examined*], in its untensed configuration, projects to a CP and attaches as a reduced relative to [$_{N'}$ *defendant*], and the repair is complete. When *shocked* arrives, it projects to VP and attaches to the existing IP, and a new trace is established in head of IP coindexed with the new main verb.

Because the existing VP structure for *examined* is left undisturbed during the repair (with the exception of the removal of the adjoined inflection), any adjuncts to the VP may be carried over without problem:

(182; UPA14) (a) The defendant carefully examined the evidence.
(b) The defendant carefully examined by the prosecutor looked nervous.

6.2.4 Lexical ambiguities

This section examines a range of structural ambiguities that arise from lexical syntactic ambiguity. The concern here is not with semantic ambiguity, though of course semantic ambiguity is nearly always involved. Rather, the focus is on syntactic ambiguity that is syntactically resolved. Because these constructions constitute about a third of the total collection, they are further divided into more manageable subgroups.

(180) *The lawyer examined with haste the defendant.

However, in the case of heavy NPs, the preference is reversed:

(181) (a) *The lawyer examined the defendant that we saw yesterday in the courthouse with haste.
(b; UPA13) The lawyer examined with haste the defendant that we saw yesterday in the courthouse.

In (181b), *examined* must be reanalysed from a reduced relative back to a main verb. NL-Soar fails to handle this repair, because the untensed *examined* cannot license a bare object NP.

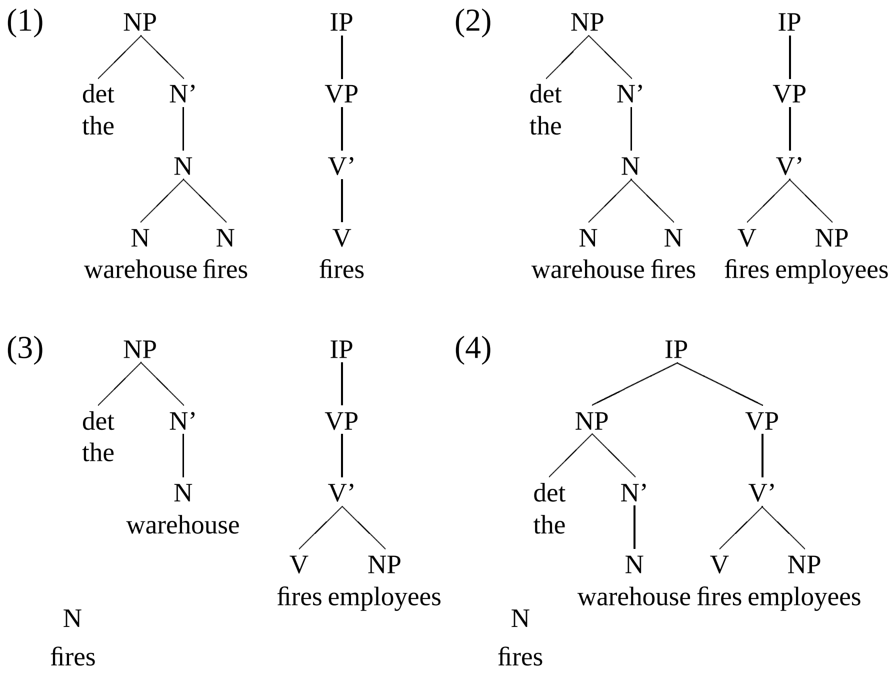


FIGURE 6.15: Repairing an unproblematic noun/verb ambiguity.

Noun/verb ambiguities

In the following constructions, the ambiguous region may be taken as a compound noun (183a) or a noun followed by a verb (183b) (Frazier & Rayner, 1982):

(183; UPA15) (a) The warehouse fires kill numerous employees each year.
(b) The warehouse fires numerous employees each year.

Figure 6.15 shows how NL-Soar can repair the compound noun structure to be consistent with the interpretation required in (183b). As described in §3.8, multiple syntactic senses of lexical items are accessed in parallel and placed in the A/R set, where they may generate their own projections. The first frame of Figure 6.15 shows the structure just before [$_{NP}$ *numerous employees*] arrives. The nominal form of *fires* has adjoined to [$_N$ *warehouse*] to form a compound noun, and the verb form has projected to a VP and tensed IP. When *employees* arrives, it attaches as the complement of [$_{V'}$ *fires*]. At this point, the two competing senses of *fires* are both incorporated into the utterance model, because each is attached to another word. This triggers a snip operator to detach *fires* as a noun, leaving the simple NP [$_{NP}$ *the warehouse*]. Next, [$_{NP}$ *the warehouse*] is attached in subject position of the IP projected from *fires* as a verb, and the repair is complete.

A similar ambiguity arises with the auxiliary *can* (Gibson, 1991):

(184; UPA16) (a) The paint can fell down the stairs.

(b) The paint can be applied easily with a new brush.

In this case, *can* forms a compound noun with *paint* and projects an IP. When *be* arrives and projects to VP, it attaches as the complement of [$_{I'}$ *can*], triggering the relevant snip, and the repair proceeds as in example (183) above.

Noun/verb ambiguities may be preceded by adjective/noun ambiguities without causing difficulty (Milne, 1982; Pritchett, 1992):

(185; UPA17) (a) The square blocks the triangle.
(b) The square blocks are red.

Figure 6.16 shows how NL-Soar repairs the main verb reading of *blocks* so that it is consistent with the NP structure required by (185b). *Square* projects both AP and NP nodes, and *blocks* projects to NP and IP (1). [$_{NP}$ *The square*] attaches as the subject of [$_{IP}$ *blocks*]. When *are* arrives, it projects to IP, and [$_{NP}$ *blocks*] attaches as its subject (2). This triggers snip operators to remove structure attached to the verb sense of *blocks*, that is, [$_{NP}$ *square*] (3). Next, [$_{AP}$ *square*] adjoins to [$_{N'}$ *blocks*] (4), triggering the removal of structure attached to the noun sense of *square*, that is, [$_{det}$ *the*] (5). Finally, [$_{det}$ *the*] is attached in specifier position of [$_{NP}$ *square blocks*], and the repair is complete (6). This is another example of a repair requiring multiple snips.

Some noun/verb ambiguities do cause difficulty. If the unproblematic ambiguity in (185) is followed by a reduced relative, the result is a garden path (Milne, 1982; Pritchett, 1992):

(186; GP18) The building blocks the sun faded are red.

Blocks is taken as the main verb (as in (185), 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 (the nominal sense of *blocks* cannot attach as the subject of *are* at this point because the NP [$_{NP}$ *blocks*] is not adjacent to [$_{IP}$ *are*]). Furthermore, there are no local inconsistencies to generate a snip, so the repair is never initiated.

Another kind of difficult noun/verb ambiguity does not involve a reduced relative (Milne, 1982):

(187; GP19) The granite rocks by the seashore with the waves.

Assuming in this case that the noun interpretation of *rocks* is pursued first, the entire string may be assigned a well-formed structure as a complex noun phrase:

[$_{NP}$ *the granite rocks* [$_{PP}$ *by* [$_{NP}$ *the seashore* [$_{PP}$ *with the waves*]]]]

Thus, no inconsistencies arise to generate the snip operator. This analysis is consistent with the intuitive feeling that the problem with (190) is not so much ungrammaticality as incompleteness. Example (165) produces a similar effect.

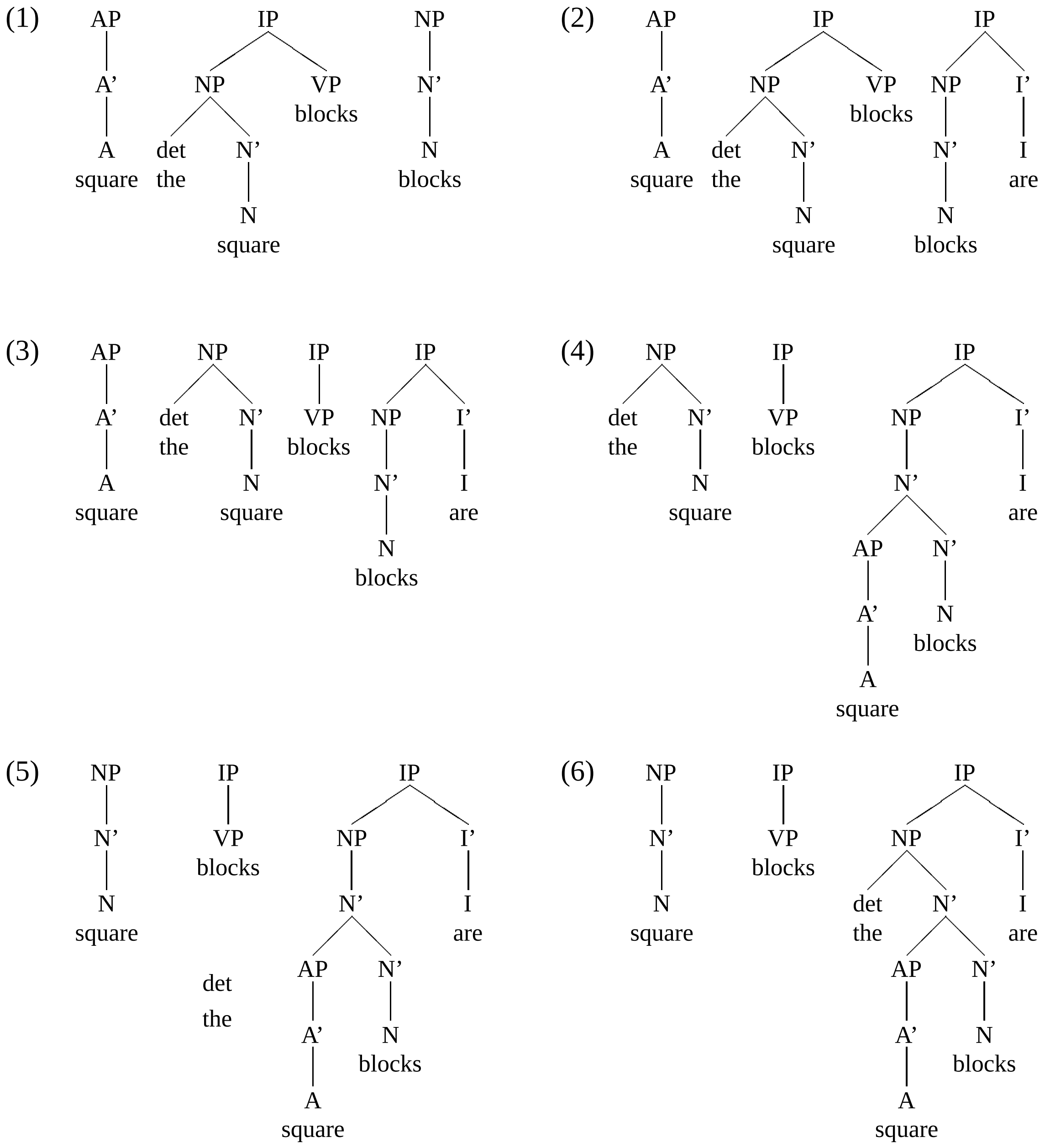

(1)
AP A' A square
IP NP det the N' N square VP blocks
NP N' N blocks
(2)
AP A' A square
IP NP det the N' N square VP blocks
IP NP N' N blocks I' I are
(3)
AP A' A square
NP det the N' N square
IP VP blocks
IP NP N' N blocks I' I are
(4)
NP det the N' N square
IP VP blocks
IP NP N' AP A' A square N' N blocks I' I are
(5)
NP N' N square
IP VP blocks
det the
IP NP N' AP A' A square N' N blocks I' I are
(6)
NP N' N square
IP VP blocks
IP NP det the N' AP A' A square N' N blocks I' I are

FIGURE 6.16: Repairing a noun/verb ambiguity preceded by an adjective/noun ambiguity.

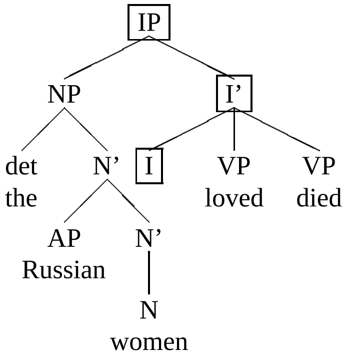


FIGURE 6.17: A garden path involving a noun/adjective ambiguity.

Noun/adjective ambiguity

This section considers a range of ambiguities involving words that can be interpreted as nouns or adjectives. We have already seen in §3.3.3 how NL-Soar could repair the basic noun/adjective ambiguity:

(188; UPA18) (a) The square is red.
(b) The square table is red.

Complement/adjective ambiguities (Pritchett, 1992) are also easily handled:

(189; UPA19) (a) I like green.
(b) I like green dragons.

In (189), *green* projects to both NP and AP. [$_{NP}$ *green*] attaches as the complement. When *dragons* arrives, it is projected to NP, and [$_{AP}$ *green*] adjoins as a modifier. This triggers the snip operator to detach the nominal sense of *green*, and [$_{NP}$ *green dragons*] attaches as the complement.

When an easily repaired ambiguity like (188) is followed by a relative clause, a garden path effect arises (Marcus, 1980):

(190; GP20) The Russian women loved died.

The Russian is first taken as an NP, then unproblematically reanalysed as *The* [$_{AP}$ *Russian*] *women*, as in (188). Next, *loved* projects to VP and tensed IP, and [$_{NP}$ *The* [$_{AP}$ *Russian*] *women*] attaches as the subject. When *died* arrives, it may attach as a VP to the I' complement, but this only succeeds in triggering a snip operator to detach [$_{VP}$ *loved*] (Figure 6.17). No further attachments are possible ([$_{VP}$ *loved*] cannot attach as a reduced relative in its tensed configuration, as discussed earlier). No further snips are generated—in particular, no snips are generated to split the nounphrase [$_{NP}$ *The Russian women*]. As a result, the repair fails.

A well-known class of ambiguous adjective-nouns do cause garden path effects, even when used in simple constructions like (188) above (Milne, 1982):

(191; GP21) The old train the children.

(Another familiar example is *The prime number few.*). Pritchett (1992) points out that the crucial distinguishing characteristic of the problematic examples is that they involve *derived nominals.* That is, the nominal sense of these words is derived from the adjective by a productive process (*the virtuous, the true, the free,* etc.). The resulting nominals appear in restricted contexts:

(192) *We saw an old yesterday.

Pritchett suggests that these words are represented in the lexicon only as adjectives, with the nominal interpretation generated by some on-line process. In NL-Soar, this means that the nominal derivation is a process of projecting an NP from the zero-level A node retrieved by lexical access. Now consider what happens in (191). Once the NP [$_{NP}$ *the* [$_{AP}$ *old*] *train*] is formed and *old* has been projected to an AP, the repair will fail. The adjective phrase [$_{AP}$ *old*] may be detached as in example (188), but no additional snips are generated to destroy the projected adjective phrase, which is necessary to make the zero-level A node available for the nominal derivation[7].

Ford, Bresnan, and Kaplan (1982) used another derived nominal to produce a garden effect with a predicate complement/subject ambiguity:

(194; GP22) The boy got fat melted.

In this case, the AP projection of *fat* can be immediately attached as the complement of *got*. When *melted* arrives, it also attaches in complement position, triggering the snip of [$_{AP}$ *fat*]. But *melted* requires the derived nominal reading of *fat*, and the zero-level A node is no longer available for the derivation. The repair thus fails. The complement/subject ambiguity itself is not the source of the problem, as demonstrated by the following sentences (Pritchett, 1992):

(195) (a) The boy got the cake.
(b) The boy got the cake baked.

Constructions like (195) can be easily handled as described earlier in the section on object/subject ambiguities.

[7] NL-Soar can still handle the unproblematic ambiguity below:

(193; UPA20) (a) The old teach the young.
(b) The old professor teaches often.

However, it does require a delay in projecting the AP until there is something to which the projection can attach (in (193b), *professor*). It is not surprising that the nominal projection is not made until the context demands it, but it is less clear exactly how this should interact with the AP projection. Nevertheless, this delay does not seriously compromise immediacy of interpretation, since no attachments to other lexical projections could be made in any event.

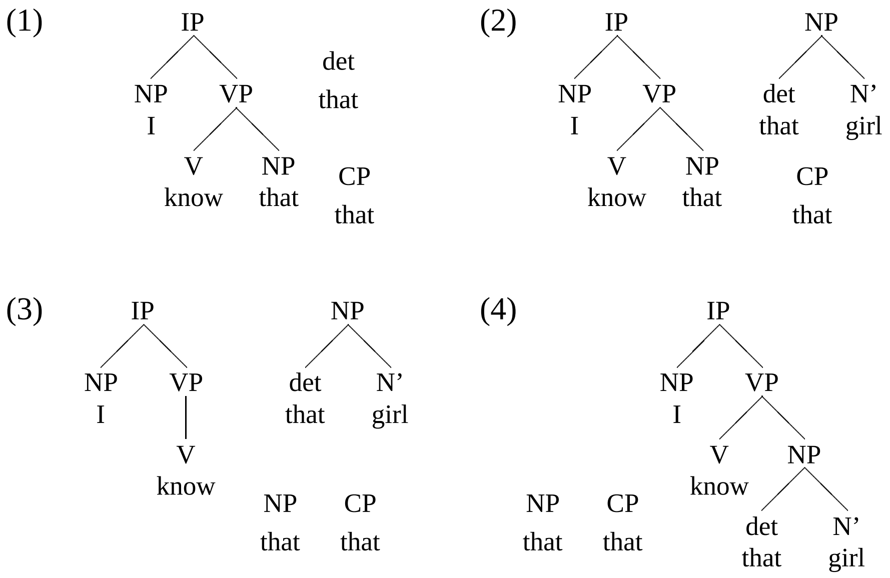


FIGURE 6.18: Repairing a pronoun/determiner ambiguity.

That ambiguity

The word *that* can play a role as a determiner, pronoun, or complementizer. Consider (196):

(196; UPA21) (a) I know that.
(b) I know that girl.

In (196a), *that* is an NP, while in (196b) *that* is a determiner. Neither sentence causes any difficulty. Figure 6.18 shows how NL-Soar handles this repair. All three syntactic senses of *that* are retrieved from the lexicon, and the projected pronominal node attaches initially as the complement of *know*. When *girl* arrives, [$_{det}$ *that*] attaches as its specifier. This triggers the snip of [$_{NP}$ *that*] from complement position, and finally [$_{NP}$ *that girl*] becomes the complement.

A similar unproblematic ambiguity involves the complementizer reading of *that*:

(197; UPA22) (a) I know that.
(b) I know that dogs should play.

(197) is handled in essentially the same manner as (196). When *should* projects to an IP, [$_{NP}$ *dogs*] attaches as its specifier. Next, [$_{IP}$ *dogs should*] attaches as the complement of [$_{CP}$ [$_{C'}$ *that*]]. This triggers the snip of [$_{NP}$ *that*], and [$_{CP}$ *that* [$_{IP}$ *dogs should*]] becomes the complement.

Surprisingly, *that* ambiguities can also produce garden path effects:

(198; GP23) Before she knew that she went to the store.

In (198), *that* is initially taken as a pronoun, then reanalysed as a complementizer for [$_{IP}$ *she went*] in the manner described above. Next, [$_{PP}$ *to the store*] attaches as the complement of [$_{V'}$ *went*]. The processing completes with the well-formed CP,

(199) [$_{CP}$ *before* [$_{IP}$ *she knew* [$_{CP}$ *that she went to the store*]]].

No snips are generated since there are no local inconsistencies, so the correct structure is not uncovered.

Garden path effects can also be produced with complementizer/determiner ambiguities (Gibson, 1991; Pritchett, 1992):

(200; GP24) I saw that white moose are ugly.

In (200) the singular/plural ambiguity of *moose* interacts with the ambiguity of *that* to cause the difficulty. *That white moose* is initially interpreted as a singular NP and attached as the complement of *saw*. *Are* arrives, projects to IP and CP (it cannot attach as the complement of [$_{CP}$ *that*] because the phrases are not adjacent), and attaches in complement position of *saw.* This triggers the snip of [$_{NP}$ *that white moose*], but [$_{NP}$ *that white moose*] cannot be attached as subject of [$_{IP}$ *are*] due to a number agreement violation. No further snips are generated and the repair fails.

Another difficult complementizer/determiner ambiguity arises when a sentence-initial *that* may be interpreted as a determiner or as a complementizer for a subject sentence (Gibson, 1991):

(201; GP25) That coffee tastes terrible surprised John.

That coffee is interpreted as an NP, which can then serve as the subject of *tastes terrible.* When *surprised* arrives, it cannot take the initial clause as a subject sentence because subject sentences must have overt complementizers:

(202) *Sarah left bothered Jill.

The only option is to project *surprised* to VP and attach it as the complement of the existing IP. This leads nowhere; in particular, it does not lead to the required snip of [$_{det}$ *that*] from [$_{NP}$ *that coffee*]. The repair thus fails.

Other lexical ambiguities

Auxiliary/main verb

Auxiliary/main verb ambiguities can give rise to garden path effects (Marcus, 1980; Kurtzman, 1985):

(203; GP26) Have the boys given gifts by their friends.

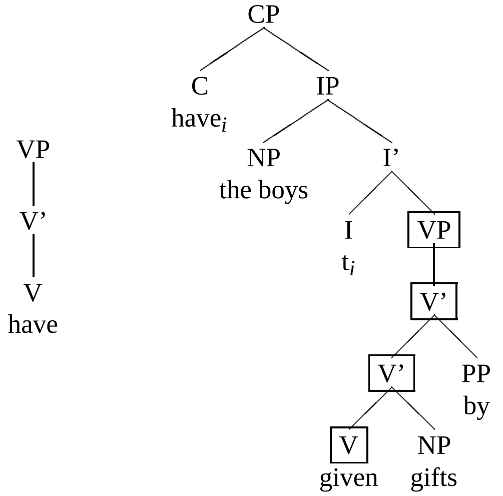


FIGURE 6.19: A main verb/auxiliary garden path.

Figure 6.19 shows how the garden path effect arises. *Have* may be interpreted as an imperative main verb (*have* heads a VP) or an auxiliary that begins a question (*have* heads a CP). The question interpretation is pursued when [$_{NP}$ *boys*] attaches in spec-IP position. *Given* projects to a VP and attaches as the I' complement, and [$_{NP}$ *gifts*] becomes the first complement of [$_{V'}$ *given*]. Next, [$_{PP}$ *by*] adjoins to [$_{V'}$ *given*]. In this configuration, *given* is missing an obligatory second complement. This may trigger a snip of the VP from [$_{CP}$ *have* [$_{IP}$ [$_{NP}$ *the boys*]]], but it does not trigger the snip required to remove [$_{NP}$ *the boys*] from the interrogative structure. Furthermore, the detached [$_{VP}$ *given*] cannot attach as the complement of [$_{VP}$ *have*] because the phrases are not adjacent. As a result, the repair fails[8].

Singular/plural

We have already seen some examples in which number ambiguity plays a role. Number ambiguity itself is not necessarily a problem (Kurtzman, 1985):

(205; UPA23) (a) The sheep seem very happy.
(b) The sheep seems very happy.

[8] This analysis also predicts the following structure to be a garden path:

(204) Have the boys go to the store.

Sentence (204) has usually been offered as an example of an unproblematic ambiguity (e.g., (Marcus, 1980; Gibson, 1991; Pritchett, 1992)). However, the only experimental evidence using precisely this structure suggests just the opposite. Using a rapid grammaticality judgment task, Kurtzman (1985) found that, at the crucial disambiguating verb (*go*), the immediate judgement of most subjects (67%) is that the string is ungrammatical. It is difficult to explain away this result as an artifact of the particular experimental paradigm, since Kurtzman reproduced a range of other well-known and accepted garden path and unproblematic contrasts. Until further evidence can be obtained, we can tentatively assume that most of Kurtzman's subjects experienced some failure of their capability to recognitionally repair the structure. The advantage of Kurtzman's technique is that it taps into the immediate on-line judgment of subjects. For this same reason, of course, it reveals no contrast between garden paths that are difficult to recover from and those that are easy.

Because there is no structural ambiguity here, there is no need to represent the ambiguity with multiple NPs. NL-Soar simply handles this ambiguity at the level of syntactic features, like nearly every other approach to natural language processing (e.g., unification grammars). The head of [$_{NP}$ *the sheep*] contains a set of syntactic features, including number, which may take on multiple values. These features are restricted as required by agreement checks. When [$_{NP}$ *the sheep*] is attached in spec-IP position of [$_{IP}$ *seems*], spec-head agreement ensures the number will be set to *singular*; when the attachment is to [$_{IP}$ *seem*], the number is set to *plural*.

Inflection marker/preposition

To may be locally ambiguous as a preposition or an inflection marker, but the ambiguity need not cause difficulty (Gibson, 1991):

(206; UPA24) (a) I opened the letter to Mary.
(b) I opened the letter to impress Mary.

Figure 6.20 shows how NL-Soar repairs from the preposition reading to the inflection reading. [$_{PP}$ *To*] initially attaches as the complement of [$_{N'}$ *letter*]. When *impress* arrives, it projects to VP and attaches as the complement of [$_{IP}$ *to*]. This triggers the snip of [$_{PP}$ *to*], and the initial clause adjoins to [$_{IP}$ *to impress Mary*], completing the repair.

6.2.5 Filler-gap ambiguities

Filler-gap sentences provide another interesting test of NL-Soar's repair mechanism, because the location of the gap is often not known with certainty. Consider (207):

(207; UPA25) (a) John found the ball$_i$ that the boy hit t$_i$.
(b) John found the ball$_i$ that the boy hit the window with t$_i$.

In (207a), the trace appears in the complement position of the verb (*the boy hit the ball*), while in (207b), the sentence continues so that the trace appears as the object of a preposition (*the boy hit the window with the ball*). Neither sentence causes difficulty. Figure 6.21 shows how the repair of the object trace is triggered. First, [$_{NP}$ *the window*] arrives and attaches as the complement of [$_{V'}$ *hit*]. This creates the familiar local inconsistency: two nodes (the trace, and the NP) occupying the same structural position. It is irrelevant that one of the nodes is a phonologically null trace. A snip is generated to remove the trace. When [$_{PP}$ *with*] adjoins to [$_{V'}$ *hit*], a new trace is generated as the object of the preposition.

A striking aspect of these filler-gap ambiguities is that they may be propagated over long distances:

(208; UPA26) (a) Who do you believe?
(b) Who do you believe Jim suspects Bob knows Pat hates?

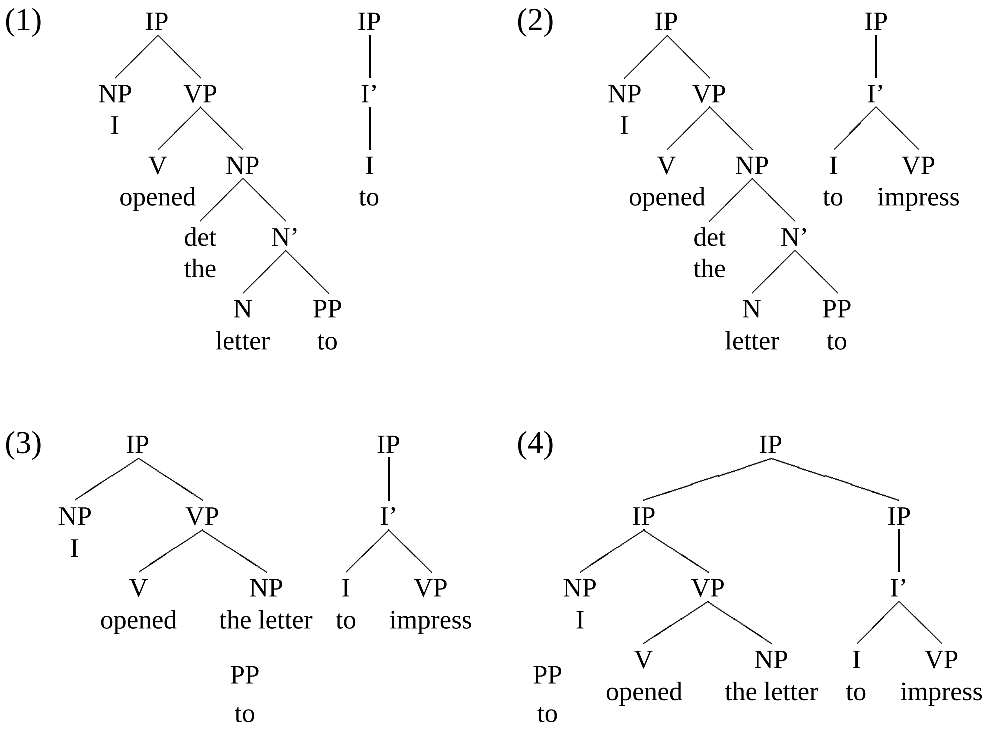


FIGURE 6.20: Repairing an inflection marker/preposition ambiguity.

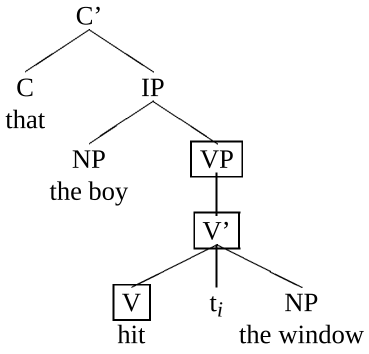


FIGURE 6.21: Triggering the repair of a filler-gap ambiguity.

An object trace is posited after each verb in (208b). As an NP arrives to fill the object slot, the trace is snipped, and a new one generated at the next verb. Thus, (208b) involves three separate repair operations.

6.2.6 Small clauses, coordination and other miscellany

Small clauses

Small clauses are subject/predicate constructions that do not involve the full IP/VP phrase structure. Small clauses may consist of just a VP, with the subject of the small clause in spec-VP. Pritchett (1992) presents the following unproblematic ambiguity involving a VP small clause:

(209; UPA27) (a) I saw her duck fly away.
(b) I saw her duck into an alleyway.

The are actually several repairs involved in (209). First is the reanalysis from [$_{VP}$ *saw* [$_{NP}$ *her*]] to [$_{VP}$ *saw* [$_{NP}$ *her duck*]], which is an instance of object/specifier ambiguity discussed in §6.2.1. If the sentence then continues as in (209a), *fly* projects to a VP and attaches as the complement of [$_{V'}$ *saw*], triggering the snip of [$_{NP}$ *her duck*]. [$_{NP}$ *her duck*] then attaches in spec-VP position, forming the small clause [$_{VP}$ [$_{NP}$ *her duck*] *fly*].

Of course *duck* is categorially ambiguous: it may be a noun or verb. Figure 6.22 shows what happens if the sentence continues as in (209b). The PP [$_{PP}$ *into*] attaches as the complement of [$_{V'}$ *duck*][9]. This triggers the snips to detach [$_{NP}$ *duck*]—one snip to remove [$_{NP}$ *her*] from [$_{NP}$ *duck*], and another to remove [$_{NP}$ *duck*] from [$_{V'}$ *saw*]. Next, [$_{NP}$ *her*] attaches as specifier of [$_{VP}$ *duck*], forming the small clause [$_{VP}$ [$_{NP}$ *her*] [$_{V'}$ *duck* [$_{PP}$ *into*]]]. Finally, the small clause attaches to [$_{V'}$ *saw*] as the complement, and the repair is complete. This sentence provides a demonstration of how NL-Soar can handle multiple kinds of structural and lexical ambiguity within one sentence.

Coordination

Coordination has received little empirical attention in psycholinguistics (though there is a fair amount of work in computational linguistics, e.g., (Kosy, 1986)), and there is potentially a huge range of interesting garden path effects waiting to be discovered. However, for the present purposes, we shall address just the most basic kind of reanalysis required by coordinate structures:

(210; UPA28) (a) I went to the mall.
(b) I went to the mall and the bookstore.

[9] This sentence is actually globally ambiguous. Attaching [$_{PP}$ *into*] to [$_{VP}$ *saw*] produces a structure corresponding to the somewhat odd interpretation *I accompanied her duck into the alleyway*.

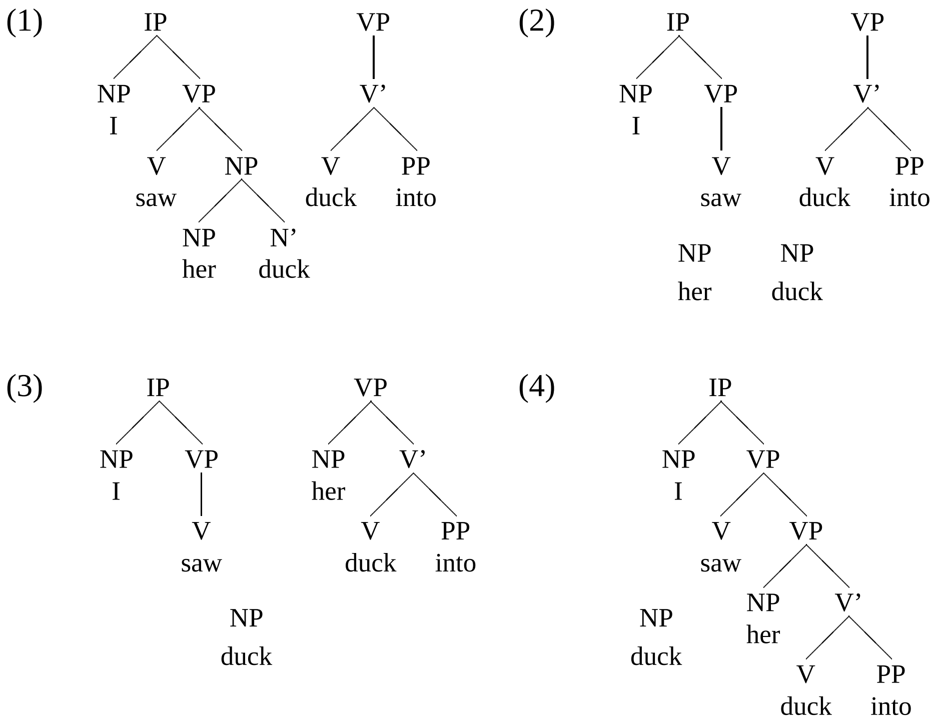


FIGURE 6.22: Repairing a VP small clause ambiguity.

Figure 6.23 shows how NL-Soar would handle the repair[10]. Initially, [$_{NP}$ *The mall*] attaches as the complement of [$_{P'}$ *to*]. *And* arrives and projects its own maximal projection. Since a conjoined phrase takes on the syntactic features of its conjuncts, the node projected by *and* does not yet have category information; this is represented in the figure by denoting the node as XP. Next [$_{XP}$ *and*] attaches in complement position, triggering a snip operator to remove [$_{NP}$ *the mall*], which can then attach as the first conjunct of [$_{XP}$ *and*]. Finally, [$_{NP}$ *the bookstore*] becomes the second conjunct, forming the conjoined phrase [$_{NP}$ [$_{NP}$ *the mall*] *and* [$_{NP}$ *the bookstore*]].

Multiple compounding

Pritchett (1992) points out that nouns may be compounded multiple times without causing difficulty:

(211; UPA29) (a) We admire their intelligence.
(b) We admire their intelligence agency policy decisions.

Compounding may be analysed as adjunction to the head of an NP. Under this analysis, the repair from a simple N to a compound N simply involves the standard adjunction operator,

[10] These coordinate structures have not yet been implemented in the system.

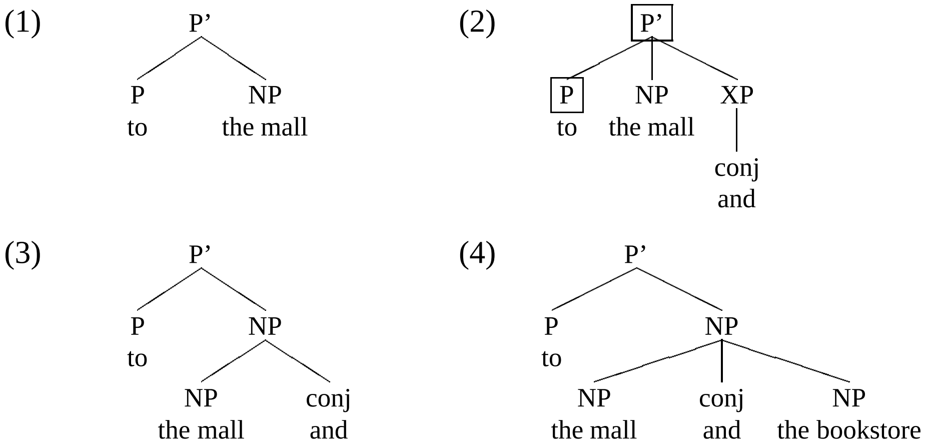


FIGURE 6.23: A repair involving coordination.

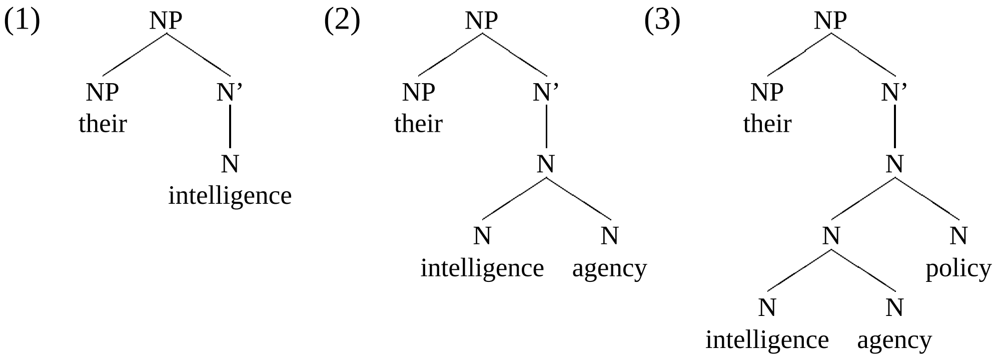


FIGURE 6.24: Multiple compounding.

which creates the additional required node. Thus, multiple compounding consists of a stream of adjunctions, as in Figure 6.24. No snip operators are required.

Semantic vs. structural ambiguity

Thus far all the structures we have considered involved local syntactic ambiguity that is resolved later in the sentence by additional syntactic information. Some kinds of local ambiguity that are resolved syntactically are purely semantic in nature, and therefore do not require repair of the utterance model. Consider the examples below:

(212; UPA30) (a) I gave the dog to Mary.
(b) I gave the dog some bones.

(213; UPA31) (a) John picked the boy for his team.
(b) John picked the boy up at school yesterday.

1. Recoverability
2. Bidirectionality
3. Independence of length
4. Distance-to-disambiguation effects
5. Independence of lexical ambiguity
6. Independence of semantic content

FIGURE 6.25: General garden path phenomena (from Chapter 2.)

In (212), [$_{NP}$ *the dog*] plays a different semantic role in the sentence depending on the syntactic structure of the second complement of *give* (whether or not is a PP). Yet, regardless of the outcome, [$_{NP}$ *the dog*] remains in the same structural position, so no repair of the utterance model is required (the situation model may be repaired as described in §3.4.3). Similarly, in (213), the meaning of *picked* changes with the arrival of the particle *up*, but the structural configuration remains the same.

Sometimes local syntactic ambiguity is resolved by later *semantic* content. In general, such conditions can give rise to *semantic garden paths* (§2.4), for example:

(214) British left waffles on Falklands.

The basic NL-Soar account of such effects is straightforward: repairing the structure online requires the recognitional generation of the appropriate snips to effect the repair. The rather specific semantic contexts of these semantic garden paths make it unlikely that the appropriate repair sequence will be available as a chunked process.

6.3 Accounting for the major qualitative phenomena

Now that we have completed the lengthy journey through the range of GP/UPA constructions, we can step back and consider how NL-Soar accounts for the six major qualitative phenomena surrounding garden path effects, summarized in Figure 6.25.

Recoverability. People can eventually recovery from garden paths through deliberate recomprehension, perhaps guided by explicit instruction. NL-Soar predicts that garden paths are recoverable because the knowledge used to resolve ambiguities is not architecturally fixed in advance. We saw in Chapter 4 how NL-Soar is able to deliberately recomprehend linguistic input to explore alternative paths at ambiguities. This capability is just what is required to recover from garden path effects (in fact, the example in Chapter 4 involved the classic main verb/reduced relative garden path).

Bidirectionality. Garden path effects may arise even when a normally unpreferred path is taken, and the preferred interpretation turns out to be correct. NL-Soar predicts this is possible because the GP effect is purely a function of the ability of the repair mechanism to transform one structure into another; the preferred/nonpreferred status of the structures is irrelevant. We saw this in GP4 and GP13.

Independence of length. Since NL-Soar's repair mechanism maps structure into structure, the length of the surface string is not the critical factor. NL-Soar predicts the existence of short garden path sentences (e.g, GP15 and GP20) as well as unproblematic ambiguities with extended distance-to-disambiguation (e.g., UPA3).

Distance-to-disambiguation effects. Although length is not the important factor, increased distance-to-disambiguation can lead to a garden path effect. In NL-Soar, this happens in just those cases where the structural characteristics of the intervening material is such that it causes loss of the critical relations from the A/R set. We saw this in the contrast between GP2, UPA1, and UPA3. Thus, NL-Soar predicts that there are not pure distance effects, but *structurally modulated* distance effects arising from syntactic interference.

Independence of lexical ambiguity. Because NL-Soar's repair maps structure to structure, lexical ambiguity is only relevant to the extent that it has structural ramifications. NL-Soar predicts that lexical ambiguity is neither necessary nor sufficient for causing garden path effects. This general prediction can be seen clearly from the results on the many examples in §6.2.4 involving lexical ambiguity.

Independence of semantic content. Semantic ambiguity need not cause a garden path effect. We saw in UPA30 and UPA31 that the utterance model need not always be repaired in the case of semantic ambiguity. Furthermore, any required situation model repair can be accomplished directly by the construction processes that keep it in correspondence with the utterance model (§3.4.3).

6.4 Summary and general discussion

Table 6.1 summarizes the predictions on the GP/UPA collection. The results are good: the theory accounts for 52 out of 57 constructions (91%). Only the theories of Pritchett (1992) and Gibson (1991) have comparable coverage, and no other *architectural* theory competes. Furthermore, no other existing theory accounts for all six qualitative phenomena (for example, NL-Soar offers the first explicit model of how people might actually recover from garden path effects). Since NL-Soar embodies the Structural Garden Path Hypothesis, the good results of the theory offer further support for this hypothesis, along with Pritchett (1992) and Gibson (1991).

Perhaps the biggest theoretical issue facing the model is the origin of the repair mechanism. What gives rise to the particular locality constraint embodied in the generator for the snip operator? Although the constraint is simple enough, and the presence of *some* constraint is well motivated computationally, the precise form of the constraint could possibly be altered slightly and still ensure efficient repair. (A form of this question can be put to nearly every other structural garden path theory.) It is interesting to note, however, that the identification of the maximal projection as the constraining locality for snip means that the theory could be mapped rather simply onto radically different structures, such as

TABLE 6.1: Summary of predictions on the UPA/GP collection.

Object/subject/specifier	UPA1 UPA2 UPA3 UPA4 UPA5 UPA6 UPA7	● ○ ● ● ● ● ●	GP1 GP2 GP3 GP4 GP5 GP6 GP7 GP8 GP9 GP10	● ● ● ● ● ● ● ● ● ●
Complement/adjunct	UPA8 UPA9 UPA10 UPA11	● ● ○ ●	GP11 GP12 GP13	○ ○ ●
Main verb/reduced relative	UPA12 UPA13 UPA14	● ○ ●	GP14 GP15 GP16 GP17	● ● ● ●
Lexical ambiguities	UPA15 UPA16 UPA17 UPA18 UPA19 UPA20 UPA21 UPA22 UPA23 UPA24	● ● ● ● ● ● ● ● ● ●	GP18 GP19 GP20 GP21 GP22 GP23 GP24 GP25 GP26	● ● ● ● ● ● ● ● ●
Filler-gap	UPA25 UPA26	● ●		
Small clauses, etc.	UPA27 UPA28 UPA29 UPA30 UPA31	● ● ● ● ●		

●= correct prediction
○= incorrect prediction

dependency trees. In a dependency grammar, maximal projections correspond directly to single lexical nodes.

Of course, there are the missed predictions to be accounted for. Three of the five incorrect predictions involve complement/adjunct ambiguities, which may help focus research for ways to improve the theory.

In summary, although there are still important issues to deal with, these results help establish the viability of the broader class of single path/simple repair models. By virtue of being embedded in the larger architecturally-grounded theory, NL-Soar also explains all the major qualitative garden path phenomena, and opens the door to an understanding of how learning modulates the recovery skills of adult comprehension.

Chapter 7

Immediacy of Interpretation and the Time Course of Comprehension

One of the primary constraints on NL-Soar is that it accomplish the functions of comprehension in an incremental, real-time fashion. As we saw in Chapter 2, immediacy of interpretation is a fundamental principle characterizing human sentence processing. Although nearly all comprehension models embody some kind of incrementality, most do not make predictions about the time course of comprehension. The READER model (Thibadeau, Just, & Carpenter, 1982) is a notable exception. Because NL-Soar is grounded in a cognitive architecture with an independently developed temporal mapping (§3.1.2), we can use NL-Soar to make chronometric predictions. The first section of this chapter examines why NL-Soar is an immediate interpreter. Next, we consider how NL-Soar satisfies the real-time constraint, by analysing the structure of the model as well as actual system behavior. Finally, the model is used to make a number of specific predictions (both qualitative and quantitative) about the time course of comprehension.

7.1 NL-soar as an immediate interpreter

In general, human comprehension processes operate immediately at all levels—syntactic, semantic, and referential (§2.2). There is no evidence of systematic delays at any level. As soon as the relevant inputs are present, the comprehension processes apply, incrementally building representations in working memory.

This is an adequate characterization of NL-Soar as well. In fact, immediacy is the most natural processing strategy for NL-Soar. It is the nature of the recognition memory in Soar that associations fire automatically whenever their relevant conditions are present. There is no central executive scheduling the processes. This is a basic property of production systems in general (Newell, 1973a), and also accounts for the immediate processing in READER (Just & Carpenter, 1987).

Consider how syntactic immediacy arises in NL-Soar. The syntactic processes are the set of available u-constructors. Whenever the conditions of a particular u-constructor proposal association are satisfied (§3.3.4), that association will fire proposing the u-constructor. Unless there is other cognitive processing that requires immediate attention (i.e., other cognitive operators are given better preferences by recognitionally available search control), the u-constructor will apply at the next decision cycle, that is, within the next 50 milliseconds or so. This immediacy holds for semantic interpretation (s-constructors) and reference resolution (resolve operators) as well.

Of course, only what is immediately available in the recognition memory (learned chunks) can be immediately applied, so NL-Soar predicts that there are limits to immediacy, and these limits are modulated by experience. The basic prediction is that the more novel and specific a particular aspect of the comprehension process, the more likely that automatic comprehension processes will fail, requiring more deliberate comprehension.

Consider what this means with respect to the three basic processes in NL-Soar (syntactic, semantic, and referential). We should expect that syntactic processing will proceed automatically most of the time for adults, since the chunked u-constructors are applicable to an unbounded variety of sentences. (Of course, this is modulo the range of failures discussed extensively in Chapters 4, 5 and 6.) Similarly, since the s-constructors build up reference-independent semantic representations based on the meanings of words, they too should applied recognitionally much of the time, though they may fail more often than the u-constructors. The referential processing is a different matter altogether. Even in a straightforward discourse, the referential level may be a source of novelty. In NL-Soar, this is also the level of processing that integrates the content into long-term memory. Some referential processing can be automatic, but if there is any new content in the discourse, then impasses must arise at some point.

It is therefore not surprising that the limits of immediacy of interpretation have been found primarily at the referential level (§2.2.3). The automatic/deliberate distinctions found in these studies maps well onto the structure of NL-Soar. However, NL-Soar does suggest that attempts to find purely static criteria for determining which aspects of comprehension are automatic are ultimately doomed to failure. Instead, the picture that emerges from NL-Soar is one in which the automatic/deliberate distinction is a function of domain knowledge and prior experience with similar language.

7.2 Satisfying the real-time constraint

Does NL-Soar in fact satisfy the real-time constraint, comprehending at an average rate of $\sim$250 ms per word? In this section we take two approaches to answering this question: analysing the structure of the model, and analysing the actual behavior of the model.

Using some rough estimates of model behavior, it is possible to determine if NL-Soar is within a plausible range. First, consider how much time is spent on recognitional utterance model and situation model construction. Each word will evoke about 1–2 u-constructors and 1–2 s-constructors. Taking 3 operators as the average total, and 50 ms per cognitive operator as the architectural constant, that means about 150 ms per word is spent on recognitional model construction. Time spent on the resolve operators is more difficult to estimate, because the amount of processing evoked by different words varies considerably. Suppose that each word evokes on average one resolve operator. This brings the total to 200 ms recognitional processing per word. Thus, this analysis suggests that the model passes the first critical test: the predicted recognitional time per word is *less* than the average time per word for human comprehenders. As discussed above, there must be some time left over for more deliberation. However, the analysis also shows that the fit is tight, leaving only an average of 50 ms per word for deliberate processing. Put another way, the analysis indicates that comprehension is about 80% recognition and 20% deliberation. Real-time comprehension clearly presses the architectural limits.

To establish with more confidence that NL-Soar meets the real-time constraint, we must determine several factors empirically:

- How often u-constructors and s-constructors impasse (the transfer rate of u-constructor/s-constructor chunks)
- How many s-constructors and u-constructors apply per word, on average
- How often resolve operators impasse (the transfer rate of reference resolution chunks)
- How many resolve operators apply per word, on average

As argued above, the transfer rate of u-constructors and s-constructors should be good, since they constitute fairly general processes. There is some empirical evidence to back this up. Figure 7.1 shows a graph depicting a learning curve on a corpus of 61 sentences (devised to test the syntactic range of the system). The horizontal axis represents the cumulative number of words comprehended; the vertical axis represents the percentage of words comprehended without impasse (averaged over a moving 24-word window). The data is from an earlier version of NL-Soar (Lehman, Lewis, & Newell, 1991; Steier, Lewis, Lehman, & Zacherl 1993) which combined utterance and situation model building into one operator. The system did not perform reference resolution. The data are still relevant because the transfer in the present system should be at least as good as the older system (see §3.7 for a discussion of the transfer properties of different comprehension operator schemes).

The system starts without any chunks that directly implement comprehension operators (though the graph starts above zero because there was some chunk transfer within the first 24 words). Without learning, the graph would be a flat line at 0%. The recognitional rate of 80-90% achieved near the end of the corpus is entirely a result of the transfer of learned chunks. It seems plausible to assume that in adult comprehension the transfer rate for utterance model and situation model construction is even higher.

The data in Figure 7.1 still leaves what is potentially the most significant factor unaddressed: the amount of time required to perform referential processing. Recently completed work by Huffman (1993) provides some rather interesting and relevant data that begins to answer this question. Huffman investigated the nature of instructable intelligent agents with Instructo-Soar, a system that takes natural language instructions to control a simulated robot in a blocks world domain. Instructo-Soar, like NL-Soar, is built within the Soar

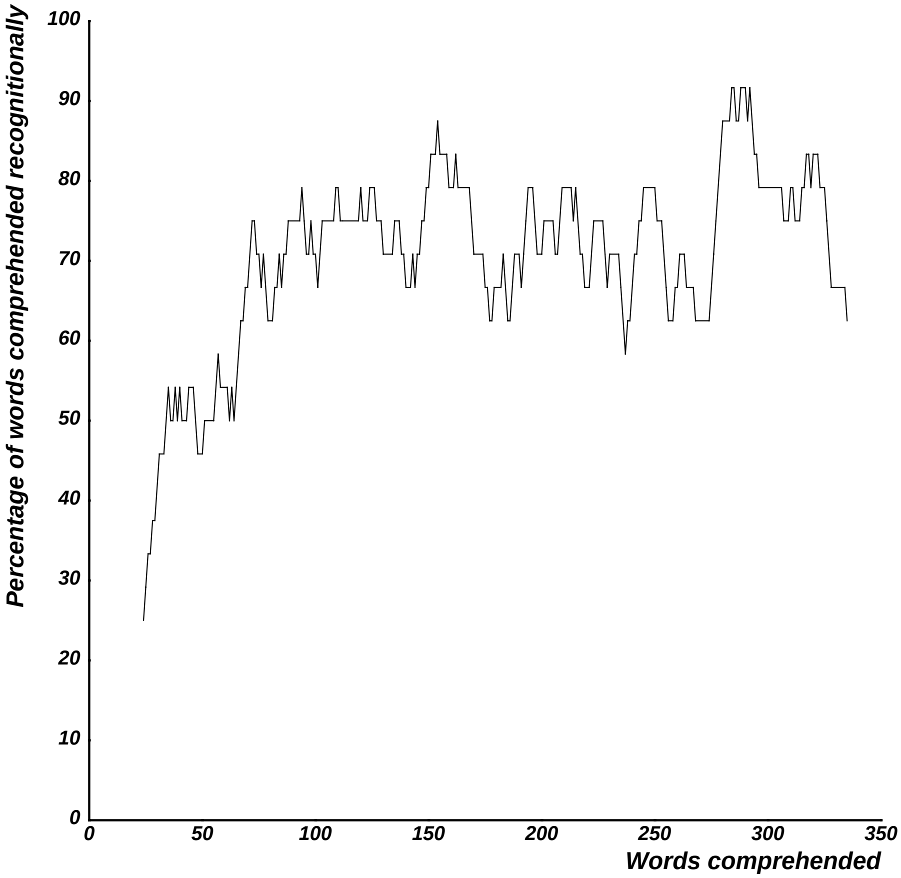


FIGURE 7.1: Efficiency increase due to chunk transfer with an older version of NL-Soar. The graph shows the percentage of words comprehended without impasse, computed over a moving 24-word window. Without learning, the graph would be a flat line at 0%.

architecture. The natural language component of Instructo-Soar is a version of NL-Soar very close to the one presented in this thesis (the only significant difference is the nature of the syntactic representation).

Instructo-Soar was run through a sequence of tasks that involved the comprehension of 94 utterances, comprised of 493 words. The utterances consisted of simple instructions and explanations such as:

(215) (a) Pick up the yellow block.
(b) Move to the grey table.
(c) The operator is finished.

A total of 566 u-constructors, 416 s-constructors, and 636 resolve operators were applied. The number of s-constructors is less than the number of words because many words (i.e., function words) did not evoke s-constructors. The total number of comprehension operators is 566 + 416 + 636 = 1, 618, or an average of 3.28 per word.

Of the 636 resolve operators, 216 led to impasses. There were 953 decision cycles spent to resolve these impasses, for an average of 4.4 decision cycles per impasse. This is just

493 words (94 utterances)

1,618 total comprehension operators
3.28 comprehension operators per word

216 impasses on resolve operators
953 decision cycles on resolve operator impasses

Total decision cycle = 1,618 + 953 = 2,571
5.22 decision cycles per word

Average time per word = 5.22 × 50 ms = 261 ms
Comprehension rate = 1000 ÷ 261 × 60 = ∼230 words per minute

FIGURE 7.2: Results from running NL-Soar in Instructo-Soar's domain. The 50 ms per operator architectural constant permits approximate zero-parameter predictions of comprehension rate.

enough time to learn to recognize one or two new aspects of the situation model (§3.5.1). Assuming that the u-constructors and s-constructors were fully chunked (they were not, of course—Instructo-Soar started with no language chunks of any kind), the total decision cycles spent was $1,618 + 953 = 2,571$, or an average of 5.22 per word. At 50 ms per cycle, that means an average of 261 ms per word, or about 230 words per minute. These results are summarized in Figure 7.2.

This comprehension rate is remarkably close to the observed rate of skilled human readers (∼240 wpm; (Just & Carpenter, 1987)). Although this is just one data point, the test is a significant one because it embeds NL-Soar in a functional task situation. Reference resolution is required for both the immediate comprehension of the text, as well as for producing a long-term memory of the instructions to be used in later behavior.

If NL-Soar appears a bit slow, it is important to realize that on Instructo-Soar's task, NL-Soar is taking the time required to produce a complete recognition memory of the content of the utterances (though the memory must be retrieved via a reconstructive process which is not guaranteed to succeed; see §3.5.3). It should be possible to speed-up NL-Soar at the cost of producing a shallower comprehension (Just & Carpenter, 1987). Nevertheless, the numbers clearly indicate that NL-Soar is operating very close to the limits of the real-time constraint.

Another factor that this analysis does not take into account is the overhead of lexical encoding and access. Although this is assumed to take place via encoding productions which can run in parallel with other comprehension operators (§3.6.3), there are still inherent data dependencies which could introduce additional latencies. It is therefore entirely possible that incorporating a more realistic lexical component and/or additional experience with other domains will reveal that NL-Soar is, in fact, too slow. In any event, the present data indicate that the deliberate processes of reference resolution need not seriously compromise the real-time behavior.

7.3 Predictions about the time course of comprehension

The previous section was about averages—establishing that NL-Soar is capable of the rapid throughput that is characteristic of skilled comprehension. However, readers do not spend the same amount of time on each word (Just & Carpenter, 1987). In this section we consider six predictions NL-Soar makes about relative processing times as a function of various aspects of the text.

To use NL-Soar to make predictions about fixation durations requires adopting the *Eye-mind Hypothesis* (Thibadeau et al., 1982; Just & Carpenter, 1987):

> (216) *The Eye-mind Hypothesis:* The eye remains fixated on a word as long as the word is being processed.

Although the Eye-mind hypothesis is generally well-supported, it is important to be explicit about this assumption, because without it no predictions can be made about relative fixation times.

The first prediction is that more time will be spent on *content* words than *function* words. Content words may evoke all three operator types (u-constructor, s-constructor, and reference operators), while function words tend to evoke only syntactic structuring operators (this was true in the Instructo-Soar corpus). This prediction is consistent with the data, but it is difficult to separate the effect from a frequency effect, since content words are not as frequent as function words.

The second prediction is that more time will be spent on *complex syntactic structures* than *simple syntactic structures*, independent of ambiguity and semantic content. The reason is that more complex structures simply require more time to build, even when they are built with a single u-constructor. Figure 7.3 shows why. If a structure contains n links such that there are inherent dependencies between the links (link x_i cannot be built until link x_{i-1} has been established), then the structure will be established with a ripple of n association firings within the application of the operator. Assuming that associations operate at the $\sim\sim$10 ms level ($\sim 3 - -30$ ms), a structure requiring $n + j$ associations will take $\sim\sim 10 \times j$ ms longer than one requiring n associations. This is a lower bound on the additional time: more complex structures are also more likely to be split into a sequence of two u-constructors. In those cases, the overhead of an additional decision cycle will be incurred.

The third prediction is that more time will be spent on *disambiguating regions* when the incorrect interpretation has been pursued. This holds for both garden path structures and unproblematic ambiguities. In garden paths, additional time will clearly be incurred because the recognitional repair fails, requiring the application of some deliberate recovery strategy. The more interesting case is recognitional repair itself. Even if the process is chunked into a single u-constructor, the sequence of associations implementing the operator is extended with the associations that accomplish the snip (or sequence of snips) and reattachments. Furthermore, repairs may take place as a sequence of two u-constructors: one u-constructor incorporating the incoming material, and a second u-constructor actually performing the repair and reattachment.

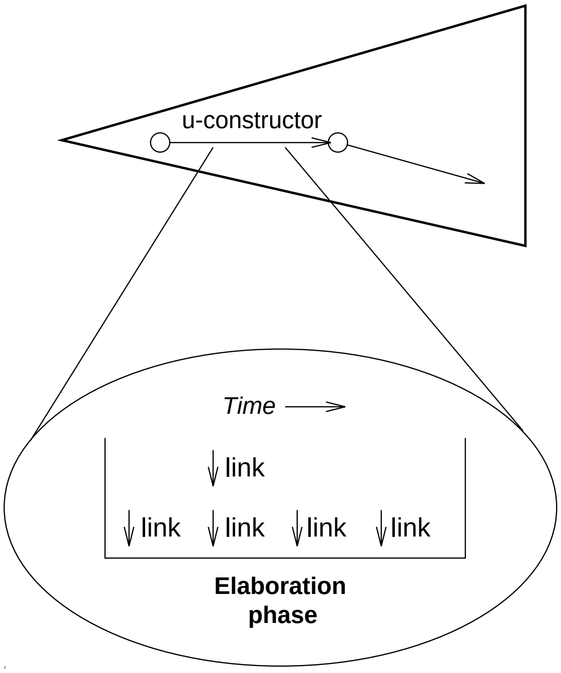


FIGURE 7.3: The elaboration phase of a u-constructor application lengthens as a function of the number of serial syntactic links.

This basic qualitative prediction is borne out in numerous studies exploring ambiguity resolution (§2.3). As an example of how the theory might be applied quantitatively, consider the contrast between the following:

(217; UPA1) (a) I forgot that Pam needed a ride.
(b) I forgot Pam needed a ride.

Sentence (217b) is an example of the familiar unproblematic subject/object ambiguity. The prediction is that the fixation time on the disambiguating region *needed* will be longer in (217b) than in (217a), because one additional process is required: the snip of [$_{NP}$ *Pam*] from [$_{V'}$ *forgot*]. At minimum the u-constructor will require an additional association, extending the operator by ∼∼10 ms. If the repair is accomplished by a separate u-constructor (which snips [$_{NP}$ *Pam*] then performs the reattachment), the time will be extended by about 50 ms. The latter account may be correct. In an eye-fixation study using material just like (217a), Ferreira and Henderson (1990) found that fixations on the disambiguating region in sentences without the overt complementizer (217b) were ∼50 ms longer than their unambiguous counterparts (217a).

The fourth prediction is that more time will be spent on *ambiguous regions* than *unambiguous regions* (MacDonald et al., 1993). Again, this extra time can take several forms. Even if the ambiguity resolution is completely chunked, the decision cycle during which the alternative operators are proposed may be extended by a series of associations that carry out the function of ambiguity resolution (Figure 3.25). These associations must fire *after* the operator proposal associations, because their conditions test for the proposed operators.

Thus, there is an inherent serial data dependency which extends the decision cycle by a few tens of milliseconds.

The fifth prediction is that there is a *filled-gap* effect (Stowe, 1986) . A filled-gap effect occurs when an overt nounphrase fills a syntactically possible gap location. Consider the following examples from Chapter 6:

(218; UPA25) (a) I saw the ball that the boy hit ␣ the window with yesterday.
(b) I saw the ball that the boy hit ␣ yesterday.

As described in §6.2.5, sentences like (218a) require a repair at the NP [$_{NP}$ *the window*]—namely, the snip of the posited object trace.

Alternatively, an impasse may arise if the ambiguity cannot be resolved by recognition. In that case, the comprehension process may be extended by a few hundred milliseconds (or even more) as knowledge in a lower problem space is brought to bear to evaluate the alternatives.

The sixth prediction, already discussed above, is that more time will be spent on *novel* aspects of the text than *familiar* aspects. This a general qualitative prediction that follows from the nature of the referential processes in NL-Soar. Novel aspects will not be immediately recognized, giving rise to impasses that build up the long-term memory of the discourse.

7.4 Summary and discussion

Figure 7.4 summarizes the predictions concerning immediacy and the time course of comprehension. These predictions derive from the structure of NL-Soar as well as analysis of system behavior. None of these predictions, including the temporal predictions, require additional assumptions beyond the basic model and the Soar architecture. The theory not only explains the fundmental immediacy principle, but also accounts for the observed distinction between automatic and deliberate processes in comprehension. Furthermore, the theory makes qualitative and quantitative predictions about the rate of comprehension and the relative time course of comprehension as a function of certain features of the utterances.

It is important to realize that these temporal predictions are essentially *zero-parameter*. Although the approximate nature of the architectural constants means the predictions are also approximate, and there is sometimes more than one possible realization of a particular function, there are no degrees of freedom in mapping the system behavior to elapsed time. In fact, NL-Soar is the first comprehension model to make zero-parameter temporal predictions. The ability to make such predictions is one of the windfalls of using a unified theory of cognition (Newell, 1990). In the case of NL-Soar, the basic architectural constants have already been established by careful analysis and application to other (non-linguistic) tasks (John, 1988; Newell, 1990; Wiesmeyer, 1992).

However, the analyses in §7.2 also raise a warning flag: NL-Soar appears to be operating at very close to the limits of the real-time constraint, and may in fact be too slow. Additional modeling and empirical work will be necessary to settle the issue conclusively.

1. Comprehension is incremental and immediate, at all levels (syntactic, semantic, referential)
2. Comprehension is mix of recognition (automatic) and deliberation; referential processing more likely to require deliberation
3. Rate of skilled comprehension is ∼230 words per minute
4. More time spent on content words than function words
5. More time spent on complex syntax than simple (tens of milliseconds)
6. More time spent on disambiguating region if wrong interpretation chosen (tens of milliseconds)
7. More time spent on filled-gaps (tens of milliseconds)
8. More time spent on ambiguous regions than unambiguous regions (tens to hundreds of milliseconds)
9. More time on novel aspects than familiar

FIGURE 7.4: Summary of NL-Soar's predictions concerning immediacy and the time course of comprehension.

This chapter effectively completes the answer to the question raised at the beginning of Chapter 1: how do people manage to comprehend so fast? It also completes our tour through the major phenomena of sentence comprehension. Before drawing general conclusions in the final chapter, the next chapter briefly explains how NL-Soar makes interesting predictions for languages other than English.

Chapter 8

Cross-linguistic Phenomena

CROSS-LINGUISTIC DATA may be brought to bear by assuming that the underlying architecture of comprehension is the same across languages. This is certainly the most natural assumption to adopt for NL-Soar, since both the Soar architecture and the additional content posited by NL-Soar make no language-specific commitments. In this chapter, we will examine NL-Soar's predictions[1] for a variety of languages, including many with structure that differs significantly from English. The first section considers parsing breakdown effects in head-final languages. The second section examines a number of cross-linguistic garden path effects and unproblematic ambiguities—some replicating effects found in English, and some involving structures with no counterpart in English. The chapter concludes with a brief summary.

8.1 Parsing breakdown on verb-final structures

Stacking three NPs in sentence-initial position leads to breakdown in the classic English center-embedded construction:

(219; PB1) The man the cat the dog chased likes cried.

Chapter 5 showed how NL-Soar accounts for this with the two-valued limitation on syntactic indices in the A/R set. However, stacking NPs is much less problematic in head-final languages, as demonstrated by the following acceptable 3-NP-initial Japanese sentence (Gibson, 1991):

(220; AE) John wa Fred ga Bill o suki da to omotteiru.
John TOP Fred NOM Bill ACC likes COMP thinks.
(John thinks that Fred likes Bill.)

[1] The NL-Soar system does not yet process the cross-linguistic examples in this chapter; the predictions are derived straightforwardly by hand.

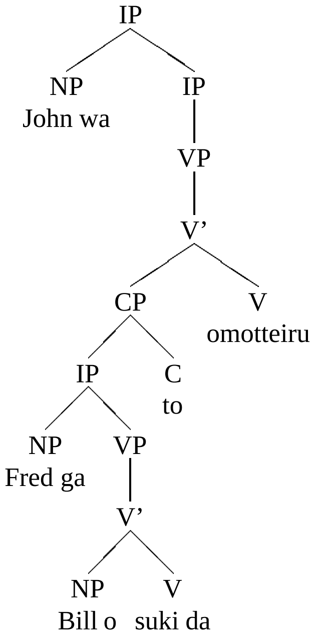


FIGURE 8.1: Phrase structure for Japanese sentence *John thinks that Fred likes Bill*, showing head-final syntax.

However, there is a crucial structural difference between (219) and (220): all three NPs in (219) occupy spec-IP (subject) position, while at most two NPs occupy spec-IP in (220)[2]. [$_{NP}$ *Bill*] occupies complement of V', as shown in Figure 8.1. Thus, no single structural relation must buffer more than two NPs in the A/R set:

RECEIVERS	spec-IP:	[$_{NP}$ *John*], [$_{NP}$ *Fred*]
	comp-V':	[$_{NP}$ *Bill*]

Even 4-NP-initial sentences may be acceptable:

(221; AE) John wa Fred ga biiruu o Dave ni ageta koto o kiita.
John TOP Fred NOM beer ACC Dave DAT gave COMP ACC heard.
(John heard that Fred gave beer to Dave.)

In (221), [$_{NP}$ *beer*] and [$_{NP}$ *Dave*] occupy first and second complement positions of [$_{V'}$ *gave*]:

RECEIVERS	spec-IP:	[$_{NP}$ *John*], [$_{NP}$ *Fred*]
	comp-V':	[$_{NP}$ *beer*]
	comp2-V':	[$_{NP}$ *Dave*]

A similar structure in German is acceptable for the same reasons (Gibson, 1991):

[2] The structural position (spec-IP vs./ adjunction) of the initial topic-marked NP in Japanese structures is a matter of debate. None of the predictions presented in this chapter depend on this distinction, though clearly examples can be devised for which the distinction is crucial.

(222; AE) Ich glaube, daß John Mary das Geschenk gegeben hat.
I believe that John Mary the present given has.
(I believe that John has given Mary the present.)

The verb-final subordinate clause in (222) stacks three NPs without causing difficulty, since the NPs are distributed across multiple structural relations (subject and complement positions).

The German counterpart to (219) does in fact cause breakdown (Gibson, 1991):

(223; PB) Der Mann den die Frau die der Hund biß sah schwam.
(The man that the woman that the dog bit saw swam.)

NL-Soar accounts for this in precisely the same manner as the English version: three NPs must be indexed simultaneously by the spec-IP relation.

RECEIVERS	spec-IP:	[$_{NP}$ *Mann*], [$_{NP}$ *Frau*], [$_{NP}$ *Hund*]

Remarkably, acceptable Japanese sentences may be found that stack *five* initial NPs[3]:

(224; AE) 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 structures may be perceived as somewhat odd, they do not cause the parsing breakdown associated with English center-embedded relatives. NL-Soar can handle structures such as (224), since no single structural relation must buffer more than two NPs:

RECEIVERS	spec-IP:	[$_{NP}$ *John*], [$_{NP}$ *Mary*]
	comp-V’:	[$_{NP}$ *Bob*]
	comp2-V’:	[$_{NP}$ *Bill*], [$_{NP}$ *Sue*]

The overt case marking exhibited in (220), (221), and (224) does not in and of itself explain the contrast between Japanese and English stacked-NPs. Even if the case markers unambiguously identify the final structural position of the NPs, there must be some way to buffer the structures until the verbs appear. Furthermore, the structural position of the NPs in unambiguous English examples such as (97) is known immediately without case marking, yet this apparently does not help the human language processor. In any event, §8.2.1 below considers examples that demonstrate that case markers in Japanese do *not* always unambiguously mark structural positions.

Not all NP-stacking is acceptable in Japanese. The following 5-NP-initial sentence does cause breakdown (Gibson, 1991):

[3] I am grateful to Brad Pritchett and John Whitman for finding this example.

(225; PB) Jon wa Mary ga Fred ga Sam ga Bill o sukida to omotteiru to sinziteiru to omotteiru.
John TOP Mary NOM Fred NOM Sam NOM Bill ACC likes COMP thinks COMP believes COMP thinks.
(John thinks that Mary believes that Fred thinks that Sam likes Bill.)

In (225), at least three NPs must be buffered on the spec-IP relation:

RECEIVERS	spec-IP:	[$_{NP}$ *Mary*], [$_{NP}$ *Fred*], [$_{NP}$ *Sam*]

In fact, it is not necessary to stack five NPs to cause difficulty in Japanese. Mazuka et al. (1989) present several examples of center-embedded structures that lead to parsing breakdown with just three initial NPs, as in the German (223) and English (219):

(226; PB) Akira ga Tosiko ga Hazime ga nakidasita toki okidasita no ni kizuita.
Akira NOM Tosiko NOM Hazime NOM started crying when got-up that noticed
(Akira noticed that Toshiko got up when Hajime started crying).

In (226), each of the NPs occupies subject position as in the English and German counterparts, requiring the A/R set to index three NPs on the spec-IP relation:

RECEIVERS	spec-IP:	[$_{NP}$ *Akira*], [$_{NP}$ *Tosiko*], [$_{NP}$ *Hazime*]

8.2 Garden paths and unproblematic ambiguities

In this section we consider a number of garden path effects and unproblematic ambiguities in Japanese, Mandarin Chinese, Hebrew, Korean, and German.

8.2.1 Japanese

Japanese case markings do not always unambiguously identify structural position. Pritchett (1991) presents several examples in which the structural position of identically case-marked NPs differs depending on the final verb[4]:

(227; UPA) (a) John ga koibito ga sinda
John NOM lover NOM died
(John's lover died)

(b) John ga Rex ga suki desu
John NOM Rex NOM fond is
(John likes Rex)

[4] Pritchett (1991) points out that, for independent grammatical reasons, these examples must be embedded to be fully acceptable. But this does not affect the ambiguity.

(228; UPA) (a) Rex wa John ga suki da
Rex TOP John NOM fond of is
(Rex likes John)

(b) Rex wa John ga mita
Rex TOP John NOM saw
(John saw Rex)

(229; UPA) (a) Rex ni John ga hanasita
Rex DAT John NOM spoke
(John spoke to Rex)

(b) John ni nihongo ga wakaru
John DAT Japanese NOM understand
(John understands Japanese)

The relevant psycholinguistic fact about these constructions is that they are unproblematic ambiguities—no matter which interpretation is required, no garden path effect arises. Because NL-Soar is head-driven, the NPs are not attached until the verb appears, so the ambiguity never actually arises. Thus, the critical question for NL-Soar is not whether the structure can be repaired, but whether the A/R set is capable of buffering the NPs such that both interpretations are possible.

Consider the *-ga -ga* ambiguity in (227). Structure (227a) is a double-subject construction in which both *-ga* marked NPs occupy spec-IP:

(230) [$_{IP}$ [$_{NP}$ John ga] [$_{IP}$ [$_{NP}$ koibito ga]]]

In contrast, (227b) is a construction in which the second *-ga* marked NP occupies object position:

(231) [$_{IP}$ [$_{NP}$ John ga] [$_{VP}$ [$_{V'}$ [$_{NP}$ Rex ga]]]]

Neither construction causes difficulty, since the NPs may be indexed by multiple relations in the A/R set:

RECEIVERS	spec-IP:	[$_{NP}$ *John*], [$_{NP}$ *Rex*]
	comp-V’:	[$_{NP}$ *Rex*]

When the disambiguating verb arrives, it is possible to build either interpretation. A similar explanation holds for (228) and (229), which exhibit different kinds of local subject-complement ambiguities. See (Pritchett, 1991) for details of the syntactic analysis.

Mazuka et al. (1989) present an interesting unproblematic Japanese construction involving main verb/relative clause ambiguity:

(232; UPA) (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 (232a), the NP NP V sequence *roozin ga kodomo o yonda* is interpreted as the main clause *The old man called the child*. In (232b), 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 (232b). Figure 8.2 shows how. 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.

Pritchett (1991) discovered one of the first known Japanese garden paths:

(233; GP) Frank ni Tom ga Guy o syookai suru to John wa iwaseta.
Frank DAT Tom NOM Guy ACC introduce COMP John TOP said CAUSE.
(John made Frank say Tom introduced Guy.)

The initial sequence through *to* is taken as a complete complementized clause (the internal structure need not concern us here; for details, see (Pritchett, 1992)):

(234) [$_{CP}$ [$_{IP}$ Frank ni Tom ga Guy o syookai suru] to]
(Tom introduced Guy to Frank.)

Next, [$_{NP}$ *John*] is encountered and left unattached, waiting for the final verb. The final verb *iwaseta* is a causative verb requiring three arguments, including an obligatory *ni*-marked causee. Only two arguments are available: [$_{NP}$ *John*] and the initial CP. The NP [$_{NP}$ *Frank ni*] must be reanalysed as an argument of [$_{V'}$ *iwaseta*]. However, the required snip within the CP is not local to the VP [$_{VP}$ *iwaseta*] which is missing the argument, so the repair fails, resulting in a garden path effect.

8.2.2 Korean

An ambiguity similar to (233) above arises in Korean as well (Pritchett, 1992):

(235; GP) Kelley-eykey Charles-ka tola o-ass-ta-ko Richie-ka malha-key hay-ss-ta.
(Richie made Kelley say Charles returned.)

As in (233), the initial sequence through *o-ass-ta-ko* may be taken as a simple clause. When the causative verb arrive, the NP [$_{NP}$ *Kelley*] must be reanalysed as the causee. The required snip is not generated and the repair fails.

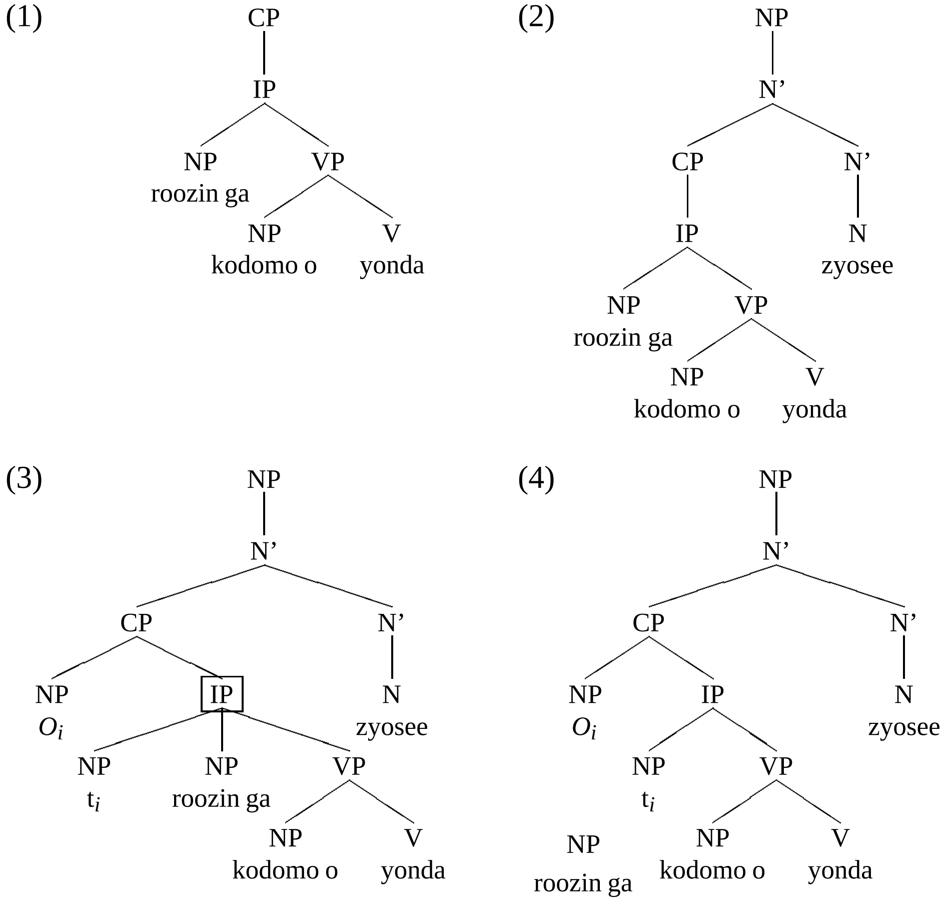


FIGURE 8.2: Repairing an unproblematic Japanese main verb/relative clause ambiguity. The initial CP is attached as a relative modifier of the incoming NP, leading to the creation of a trace in spec-IP. This triggers the snip of the misanalysed NP, making it available to serve as subject of the main clause.

8.2.3 Mandarin Chinese

Recall the unproblematic subject/object ambiguity in English:

(236; UPA1) (a) I forgot John.
(b) I forgot John went to Princeton.

The same kind of unproblematic ambiguity arises in Mandarin as well (Pritchett, 1992):

(237; UPA) (a) Wo wang le Zhangsan.
I forget PERF Zhangsan.
(I forgot Zhangsan.)

(b) Wo wang le Zhangsan yao qu.
I forget PERF Zhangsan will go.
(I forgot Zhangsan would go.)

NL-Soar repairs the Mandarin structure just as it does the English counterpart—the incoming clause attaches as the complement of [$_{V'}$ *forgot*], triggering the local snip of [$_{NP}$ *Zhangsan*]. In the same way, NL-Soar correctly predicts the following subject/object garden path (Gorrell, 1991):

(238; GP) Zhangsan yi du shu jiu diao le.
Zhangsan as-soon-as read book then fall PERF.
(As soon as Zhangsan read the book fell.)

[$_{NP}$ *shu*] attaches as the complement of [$_{V'}$ *du*], and the initial clause adjoins to the incoming [$_{IP}$ *fell*]. However, the snip of [$_{NP}$ *shu*] is not generated, as in the English case ((149), §6.1).

8.2.4 Hebrew

The subject/object garden path arises in Hebrew as well (Pritchett, 1992):

(239; GP) Axrey she-shatiti maim hitgalu be-b'er.
After COMP drank-ls water were-found in the well.
(After I drank water was found in the well.)

The explanation is the same as in the Mandarin and English examples: the local snip is not generated to remove [$_{NP}$ *maim*] from complement position.

8.2.5 German

Crocker (1990) presents an example of a garden path in German involving an object/object ambiguity:

(240; GP) daß der Entdecker von Amerika erst im 18 Jahrhundert erfahren hat
that the discoverer of America first in 18th century learned of has
(that the discoverer originally learned of America in the 18th century)

The PP [$_{PP}$ *von Amerika*] is initially taken as the complement of [$_{N'}$ *Entdecker*] to form the NP *the discoverer of America*:

(241) [$_{NP}$ der [$_{N'}$ Entdecker [$_{PP}$ von Amerika]]]

When [$_{IP}$ *erfahren hat*] arrives, [$_{PP}$ *im 18 Jahrhundert*] is adjoined as a modifier. *Erfahren* requires two arguments, but only one is available: [$_{NP}$ *der Entdecker von Amerika*]. The PP [$_{PP}$ *von Amerika*] must be reanalysed as object of *erfahren*. However, the required snip

Table 8.1: Summary of NL-Soar's cross-linguistic coverage.

German	AE	Acceptable 3-NP subordinate clause	(222)
	PB	Difficult 3-NP center-embedded	(223)
	GP	Object/object garden path	(240)
Hebrew	GP	Subject/object garden path	(239)
Japanese	AE	Acceptable 3-NP-initial	(220)
	AE	Acceptable 4-NP-initial	(221)
	AE	Acceptable 5-NP-initial	(224)
	PB	Difficult 5-NP-initial	(225)
	PB	Difficult 3-NP-initial	(226)
	UPA	Unproblematic *-ga -ga* ambiguity	(227)
	UPA	Unproblematic *-wa -ga* ambiguity	(228)
	UPA	Unproblematic *-ni -ga* ambiguity	(229)
	UPA	Unproblematic main verb/relative ambiguity	(232)
	GP	Dative/CAUSEE garden path	(233)
Korean	GP	Dative/CAUSEE garden path	(235)
Mandarin	UPA	Unproblematic subject/object ambiguity	(237)
	GP	Subject/object garden path	(238)

is not generated, because the complement relation is not local to the VP with the missing argument. As a result, the repair fails[5].

8.3 Summary

Table 8.1 summarizes the examples analysed in this chapter. Although the range of constructions is small compared to the substantial English collection addressed in Chapters 5 and 6, the variety is great enough to establish NL-Soar as a viable candidate for a universal comprehension theory.

NL-Soar's predictions about NP-stacking in head-final languages may be counter-intuitive to the native English speaker, but the contrasts among the stacked-NP constructions (particularly (220), (224), (225), and (226)) provide additional support for the structure of the A/R set. These head-final structures are important because they permit testing the theory in ways that are simply not possible with head-initial languages.

NL-Soar also establishes that head-driven, bottom-up parsing need not predict undue difficulty in head-final languages, as is sometimes supposed (Frazier, 1987). Predictions of difficulty must be made with respect to precisely articulated assumptions about the underlying mechanisms supporting comprehension. The assertion that buffering additional

[5] A similar ambiguity involving an object/adjunct ambiguity (Crocker, 1990) does not cause a garden path effect. Pritchett (1992) explains this by asserting that adjuncts need not be immediately attached in head-driven parsing, but it is not clear how such a strategy should be realized in NL-Soar.

NPs always increases difficulty carries with it implicit assumptions about the structure of short-term linguistic memory. NL-Soar clearly demonstrates alternative mechanisms are possible which do not necessarily predict overload with stacked-NPs.

The predictions for the garden path and unproblematic ambiguities provide additional support for NL-Soar's repair mechanism. It is important to establish that structures identical to GP structures in English also yield garden path effects in other languages. Just as importantly, the success of the theory on structures that have no counterpart in English increases confidence in limited repair as a universal mechanism. We also saw another clear example of the structure-sensitivity of the theory: the main-verb/relative clause ambiguity in Japanese. On the surface, this ambiguity is quite similar to the English garden path, yet NL-Soar correctly predicts that the structure can be unproblematically repaired.

This chapter completes the detailed application of NL-Soar to psycholinguistic data. In the next and final chapter, we step back and evaluate NL-Soar as a comprehensive model of sentence processing, and place it the context of some closely related theories.

Chapter 9

General Discussion and Conclusion

There are more things in an architecture, Horatio,
than are dreamt of in your theorizing.
— Allen Newell

NOW THAT THE PHENOMENA, theory, and predictions have been described in depth, we can step back and evaluate NL-Soar as an integrated psycholinguistic model, and situate it within the context of other sentence processing theories. This chapter first presents a summary of the model and its predictions, followed by a discussion of the theoretical and empirical consequences of embedding NL-Soar within the Soar architecture. The next section explores several theoretical issues (e.g., individual differences) that did not receive adequate attention in the previous chapters. Next, some closely related theories are discussed and compared with NL-Soar. The thesis closes with a look at challenging areas for future work, and a brief conclusion.

9.1 Summary: the model and its predictions

NL-Soar is first and foremost a functional model that posits computational mechanisms for realizing the task of comprehension. The model is based on an independently developed theory of the cognitive architecture, which specifies the basic control structure, memory structures, processing primitives, and learning mechanism. Table 9.1 summarizes the fundamental principles of NL-Soar, all of which are described in detail in Chapter 3.

Table 9.2 summarize the predictions of NL-Soar, as described in Chapters 4–8. All of these predictions derive from interactions of the basic principles of the model, and basic principles of Soar. Many of the predictions are novel (these are marked in the table), in that NL-Soar is the first theory to clearly make the prediction.

In addition to these general predictions, NL-Soar provides a detailed account of a wide range of garden path effects, unproblematic ambiguities, parsing breakdown, and acceptable embeddings. The model has been applied to a collection of 118 different constructions

TABLE 9.1: The basic characteristics of NL-Soar.

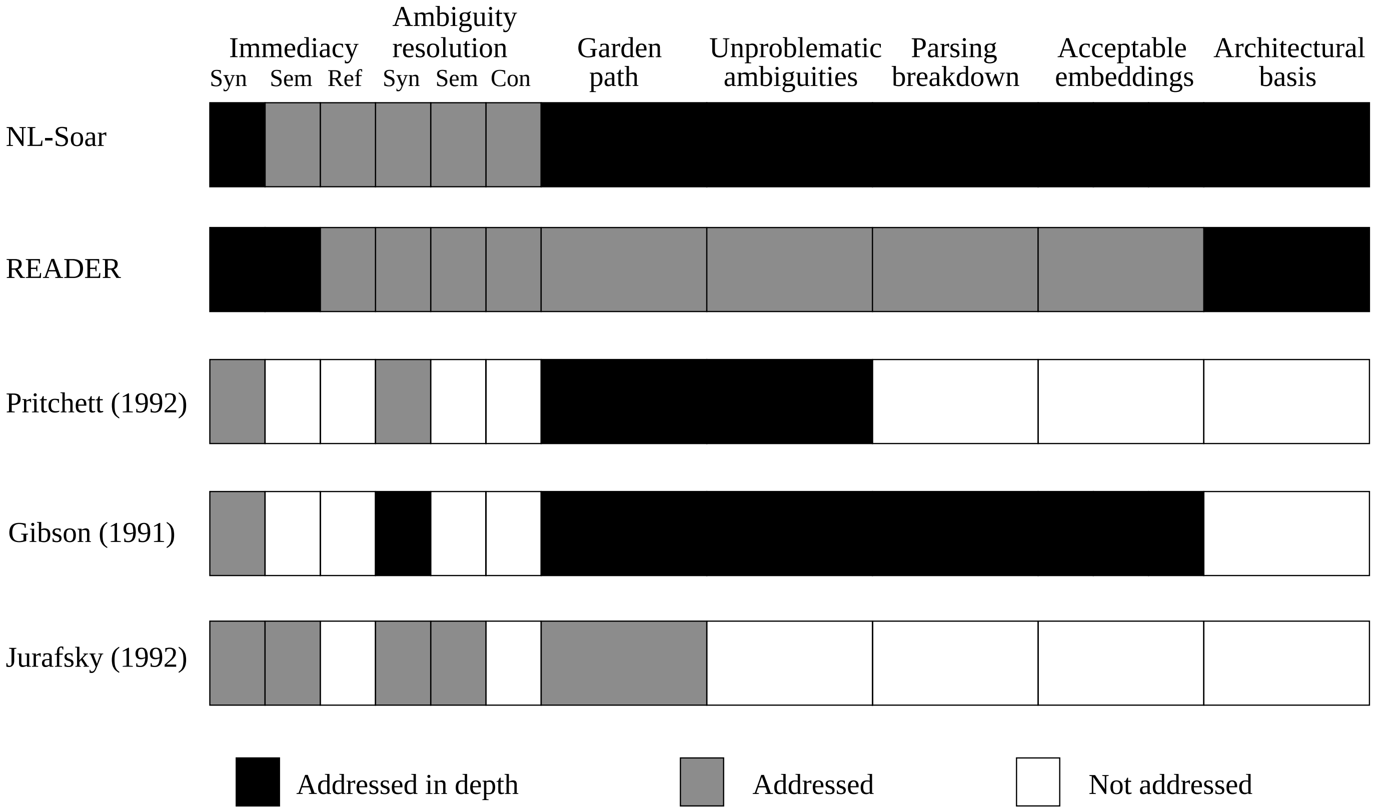
1. Comprehension operators (incremental u-constructors, s-constructors, resolve operators)
2. Comprehension as a continually improving mix of deliberate, recognitional behavior
3. Models representation of syntax, meaning, and reference
4. Limited syntactic index for utterance model
5. Context-independent, parallel lexical access
6. Head-driven, constraint-based construction of utterance model
7. Simple destructive repair mechanism
8. Reference resolution as recognition of semantic descriptions;
   reconstructive memory of discourse
9. Open, mixed parallel/serial control structure

representing these phenomena (including 17 cross-linguistic examples), with a success rate of about 92% (108 correct predictions). The results are summarized in Table 9.3.

Figure 9.1 provides a qualitative comparison of NL-Soar to some other related theories, evaluating each model with respect to these particular sentence-level phenomena. Although this comparison does not take into account all of the considerations important to the other theories (e.g., individual differences), it should be clear from Chapter 2 that these phenomena form an important core to be addressed by any model of sentence comprehension.

9.2 The role of the architecture

NL-Soar is deeply shaped by the Soar architecture. By now it should be clear that Soar is more than just an implementation language for NL-Soar. All of the fundamental principles of Soar have theoretical and ultimately empirical consequences for the model. A few examples will help further clarify the point. Consider NL-Soar's control structure—it *is* the control structure of Soar. This leads directly to the open nature of ambiguity resolution, as well as the flexibility for error recovery (§4.2). Soar's recognition memory and control structure together lead to several of the interesting limitations of ambiguity resolution (§4.1), and the distinction between automatic and deliberate processes (§7.1). The continuous learning mechanism of Soar leads to the prediction that various aspects of comprehension can be modulated by experience (§4.2,§7.1). It also provides the basic mechanism for assembling the required recognitional comprehension operators (§3.3.4). The temporal mapping of Soar is what permits the zero-parameter chronometric predictions of comprehension rates and the relative time course of comprehension (§7.2,§7.3). The nature of chunking leads to a reconstructive memory of comprehended content. The concern for efficiency in problem space search and knowledge search (recognition match) motivates the limited repair mechanism (§3.3.3).



	Immediacy			Ambiguity resolution			Garden path	Unproblematic ambiguities	Parsing breakdown	Acceptable embeddings	Architectural basis
	Syn	Sem	Ref	Syn	Sem	Con					
NL-Soar	■	▨	▨	▨	▨	▨	■	■	■	■	■
READER	■	■	▨	▨	▨	▨	▨	▨	▨	▨	■
Pritchett (1992)	▨	□	□	▨	□	□	■	■	□	□	□
Gibson (1991)	▨	□	□	■	□	□	■	■	■	■	□
Jurafsky (1992)	▨	▨	□	▨	▨	□	▨	□	□	□	□

■ Addressed in depth ▨ Addressed □ Not addressed

FIGURE 9.1: Qualitative comparison of NL-Soar theory to existing theories.

TABLE 9.2: Summary of NL-Soar's predictions.

AR	1. On-line ambiguity resolution potentially open to any knowledge source
	2. Ability to detect ambiguity limited by syntactic working memory
	3. Recency effect beyond 2 attach. sites for same relation, *ceteris paribus*
	4. Under press of time, relevant knowledge may not be available to resolve*
	5. Linguistic *Einstellung* may occur (masking of deeper knowledge)*
	6. Certain ambiguities (e.g., subject/object) not immediately detected
	7. Ambiguity resolution behavior modulated by learning*
PB/AE	8. Parsing breakdown independent of ambiguity
	9. Insufficiency of embedding depth alone to cause parsing breakdown
	10. Sharp drop in acceptability at two center-embeddings
	11. Parsing breakdown independent of length
	12. Parsing breakdown independent of short-term item memory
	13. Limited effect of explicit instruction and training on parsing breakdown
	14. Potential for effect of semantic content (assuming item memory)
	15. Stacking NPs sometimes acceptable in head-final languages
GP/UPA	16. Garden path effects a function of context, experience, and structure*
	17. Garden path effects recovered from by careful reprocessing*
	18. Garden path effects bidirectional
	19. Garden path effects largely independent of length
	20. Structurally-modulated distance-to-disambiguation effects can arise*
	21. Lexical ambiguity neither necessary nor sufficient for garden path
	22. Semantic ambiguity not sufficient for garden path
Imm/TC	23. Comprehension incremental and immediate at all levels
	24. Comprehension is mix of recognition (automatic) and deliberation; referential processing more likely to require deliberation*
	25. Rate of skilled comprehension is ∼230 words per minute*
	26. More time on content words than function words
	27. More time on complex syntax than simple (tens of ms*)
	28. More time on disambig. region when wrong path chosen (tens of ms*)
	29. More time on filled-gaps (tens of ms*)
	30. More time on ambig. regions than unambig. (tens to hundreds of ms*)
	31. More time on novel aspects than familiar

AR = structural ambiguity resolution, PB = parsing breakdown, AE = acceptable embeddings
GP = garden path effects, UPA = unproblematic ambiguities
Imm = immediacy of interpretation, TC = time course of comprehension

*Novel prediction

TABLE 9.3: Summary of NL-Soar's predictions on the GP/UPA/PB/AE collections (including the 17 cross-linguistic examples).

COLLECTION TYPE	NUMBER OF ITEMS	CORRECT PREDICTIONS	% CORRECT
Unproblemtic ambiguities	36	33	92
Garden paths	31	29	94
Parsing breakdown	20	17	85
Acceptable embeddings	31	29	94
TOTAL	118	108	92

The richness of architecturally-based computational theories is also reflected in the variety of ways that such theories may be used (Newell, 1990). The theoretical derivations in Chapters 4–8 exhibit this variety. A number of important qualitative predictions were derived from the basic structure of NL-Soar and Soar. Many detailed predictions were verified by system runs (the cross-linguistic predictions were made by hand-simulation). Approximate temporal predictions were generated in several ways: directly examining the structure of the system, making estimates of model behavior, and using traces of actual system behavior.

9.3 Some theoretical issues

Several important theoretical issues, such as modularity (§4.4) and parallelism (§3.6.3) have already been dealt with. This section examines a few other relevant topics.

9.3.1 The A/R set and the magic number two

The structure of the A/R set and the limit of two nodes per structural index plays a key role in the predictions on parsing breakdown (Chapter 5). The motivation for this structure goes beyond its ability to correctly predict the difficulty of center-embeddings. The following summarizes the functional, psychological, and computational foundations for the A/R set.

Functional motivation for A/R set

The basic structure of the A/R set is designed to effectively comprehend sentences (§3.3.3). It indexes the partial syntactic structure in a way that makes generating new structural relations and interpreting existing ones a simple match process. Furthermore, the particular set of discriminating relations are not arbitrary, but are derived directly from X-bar syntax.

Interference in short-term memory

As described in §5.1, the A/R set can be characterized as a kind of syntactic interference theory. Although content-dependent interference has not been important in theories of linguistic short-term memory, it has played an important role in classic short-term memory work emphasizing phonological working memory. (Interference is also important in long-term memory work, though that will not concern us here). Three important principles to emerge from this work are *decay*, *chunking*, and *interference* (Baddeley, 1990; Simon & Zhang, 1985). The models of Baddeley and Simon and Zhang both assume a store that holds phonological information which decays in about two seconds, unless the memory is refreshed, through overt or covert rehearsal. The short-term span is thus whatever subjects can articulate in about two seconds. Interference shows up as the *phonological similarity* effect. The span for phonologically similar items is greatly reduced (Baddeley, 1966; Simon & Zhang, 1985). For example, Baddeley (1966) found that immediate serial recall of five one-syllable words dropped from about 80% accuracy to 10% accuracy when phonologically similar word were used (*map, can, mat, cap, mad, man*) in contrast to phonologicaly dissimilar words. Simon and Zhang (1985) conducted the most extreme possible test of phonological similarity, using sequences of Chinese character homophones. For example, they used a sequence of orthographically and semantically distinct characters all pronounced “gong”. Span for characters dropped from six to seven for nonhomophonic sequences to *two* to *three* for homophonic sequences.

We can only speculate at this point about the relationship between the classic theories of short-term memory and the structure of the A/R set. However, a consistent picture that emerges is one that characterizes human short-term memory in terms of indexing and discrimination (using phonological features in the case of phonological memory, syntactic features in the case of syntactic memory), with severe limitations on the ability to represent indiscriminable contents. There may even be some indication of commonality across the two domains in terms of what that severe limit is: the magic number two of the A/R set and the 2–3 span for pure homophones in the phonological case. In general, however, there are no reasons to expect that the constants associated with decay rate and interference effect should be the same across domains (the decay rates are certainly different for visual and auditory short-term stores (Baddeley, 1990)).

Computational foundation

As mentioned in §3.3.3, there is good computational reason to expect limits on undiscriminated sets. The work on the recognition match in Soar identifies undiscriminated sets as the primary source of potentially exponential match expense (Tambe et al., 1990). One of the key methods of eliminating unwanted combinatorics in the match is to completely eliminate multiply-valued attributes in working memory. The limit of two in the A/R set comes close to this uni-attribute scheme.

The magic number two in sentence processing

The empirical motivation for the constant two is fairly broad just within the area of linguistic processing addressed in this thesis. Apart from simply predicting the difficulty on center embedding, it captures a wide range of contrasts between acceptable and difficult embeddings (Chapter 5). We saw in Chapter 4 that it also predicts a severe limitation on the ability to detect ambiguity. It leads to structure-modulated length effects in garden path phenomena that otherwise have no explanation (Chapter 6). Finally, it captures some interesting acceptability contrasts in NP-stacking in head-final languages (Chapter 8).

It is not surprising that the number two shows up in other psycholinguistic theories as well: Kimball's principle of two sentences (Kimball, 1973), A-over-A Early Closure (Church, 1980), Gibson's modified closure principle (Gibson, 1991), and various architectural theories of parsing breakdown (§2.5.4) all involve two-ness as a key feature.

9.3.2 Individual differences

Individual differences can potentially show up in every aspect of NL-Soar that is realized by associations in long-term memory. For example, ambiguity resolution, referential processing, and even the ability to recognize complex syntactic constructions can all be modulated by learning in NL-Soar, and thus are potential loci of individual differences. The fact that NL-Soar predicts such differences, and provides the explicit mechanisms that explain the nature of the differences and how they emerge, is one of the novel strengths of the model.

Such differences may be characterized as *knowledge-based* differences, in contrast to *architectural* differences. Any theory of individual differences must provide some degree of variability. For NL-Soar, that variability is in the content of the recognition memory, which determines what aspects of the comprehension skill are automatic (chunked), and what aspects are deliberate. NL-Soar does not provide variability in the underlying architecture.

However, recognition memory has a considerable amount of inertia to it—once a massive body of associations has been built up for a particular skill (such as for comprehension), that body of associations takes on architectural characteristics, in the sense that the architecture is what is relatively fixed about cognition. (In fact, we saw in §4.4 how the comprehension skill of NL-Soar exhibits various aspects of modularity, a putatively architectural concept.) Even though the comprehension skill in NL-Soar may be *modulated* by experience, it certainly cannot be fundamentally changed on any short time scale. The change must be rather slow, because any local processing can only add a fixed number of chunks that is tiny compared to the total amount of associations devoted to comprehension. What this means with respect to individual differences is that the distinction between knowledge-based and architecturally-based differences becomes somewhat blurred.

It is nevertheless still possible to consider how NL-Soar might be changed to introduce pure structural variability. The obvious candidate for variability is the magic constant two of the A/R set. Assuming that this constant reflects some rather fundamental technological limitation (see the discussion above), it would not be unreasonable to speculate that this limitation may show some variability across individuals. Such variability would be more along the lines of the variable working memory capacity of CC READER (Just & Carpenter, 1992). Although developing such a model is beyond the scope of this thesis, one clear prediction would be that groups of subjects of similar working memory capacity will still exhibit the same *contrasts* on the parsing breakdown collection that the two-valued A/R set produced. For example, object relatives will still be more difficult than subject relatives, center-embedding more difficult than right-branching, and so on—even though performance on identical embedding levels may differ across subjects.

9.3.3 Grammar and parser

This thesis did not directly address the relationship of grammar and parser, but because NL-Soar is functionally complete and does bring to bear grammatical knowledge, it is possible to derive some answers from the theory.

NL-Soar reflects rather clearly a competence/performance distinction. The grammatical knowledge in the system may be given a characterization that is independent of the particular form in which that knowledge is held. Furthermore, it is easy to see how NL-Soar exhibits various performance effects that make it fall short of perfectly realizing the competence grammar. In Chapter 2.3 on ambiguity resolution, this was called a failure of the symbol level to implement the knowledge level—more general terminology for talking about the same distinction. The limitations of ambiguity resolution, garden path effects, and so on are all examples of NL-Soar's mechanisms failing to perfectly bring to bear the grammatical knowledge which is nevertheless represented in its various problem spaces.

More can be said about the form of grammatical knowledge in NL-Soar: the grammar is in a *compiled* form when it is brought to bear recognitionally. However, the compiled aspects are still independently represented in lower spaces, so there is considerable redundancy in the system at any given point in time. As pointed out in Chapter 3, the view of grammar as consisting of a small number of interacting principles fits well within this structure. The richness of the interactions among the grammatical constraints is the engine that builds up (via chunking) the large set of efficient parsing associations in recognition memory.

Although the choice of government and binding theory was not particularly motivated by psycholinguistic data, it is clear that the choice of grammar has implications for NL-Soar's predictions. The NL-Soar model can be partially abstracted away from the grammar, but the abstracted theory cannot make empirical predictions. The predictions depend on the ontology of syntactic relations and the precise structures assigned by the grammar. For this reason, GB has undeniably played a role in the success of the theory. Of course, this does not mean that some other theory could not have done equally well; such comparisons are potentially fruitful but are beyond the scope of this thesis. In any event, some form of grammar must be adopted for functional reasons, and it is the nature of NL-Soar that the choice will have empirical consequences.

There is one important constraint NL-Soar does place on grammatical theory: it must incorporate locality constraints and break long-distance dependencies into a series of local

TABLE 9.4: Varieties of learning in NL-Soar.

1. New operator creation (newly created u-constructors, s-constructors)
2. Search control (search control for ambiguity resolution)
3. Learning from external input (reference resolution chunks encode recognition memory of new content)
4. Operator application (for new comprehension operators)
5. Learning from failure (from constraint checks on link operators; also from careful reprocessing of input triggered by semantic anomaly)

relations. GB clearly satisfies this constraint with its chain formation subject to subjacency. To see why NL-Soar requires locality, consider the severe extraction violations in (242):

(242) *Who$_i$ does Phineas know a boy who hates the man who saw t$_i$?

The severe limitations of the A/R set means that the partial syntactic structure may not be available to establish the necessary relations. In (242), NL-Soar is unable to establish the long distance relation between the object of *saw* and the initial *who*. By the time the final embedded clause is encountered, the intervening CPs will have pushed the matrix-level CP from the A/R set. Thus, the crucial spec-CP position occupied by *who* will not be available as an antecedent for the object trace:

ASSIGNERS	spec-CP:	[$_{CP}$ *who hates*], [$_{CP}$ *who saw*]

Thus, syntactic interference effects in short-term memory may explain why there is a requirement for *some* locality constraint on grammatical representation. But at present it does not seem possible to derive the precise form of subjacency or any other empirically adequate locality constraint.

9.3.4 Learning

Learning permeates every aspect of NL-Soar. One of the central results of Soar research is that many varieties of learning may emerge from a single chunking mechanism working over different kinds of problem solving impasses (Newell, 1990; Steier et al., 1987). NL-Soar itself exhibits several kinds of learning, summarized in Figure 9.4.

Chunking in NL-Soar also raises a number of serious issues, some of which remain to be resolved. Of course, there is the question of whether chunking is adequate to the task of language acquisition. The natural hypothesis of NL-Soar is that the lexicon and the language-specific aspects of grammar are all acquired via chunking (Newell, 1990). Ultimately, such a theory will place additional constraints on the nature of the lower problem spaces that are now simply posited to hold the grammatical knowledge.

Another related fundamental issue is the masking effect. We saw in Chapter 4 that chunking can produce a recognition memory that may mask knowledge in lower spaces. This is clearly an interesting psycholinguistic prediction, but it raises concerns about whether this limitation will in fact prove to be too severe. It may not be possible to settle the issue without understanding the basic structure of acquisition in NL-Soar.

9.4 Related theoretical work

This section compares NL-Soar with a few closely related theories: the production-system model of Just and Carpenter, and the principle-based parsing theories of Pritchett and Gibson.

9.4.1 READER and CC READER

The NL-Soar model is theoretically closest to the READER and CC READER production system models of comprehension (Thibadeau et al., 1982; Just & Carpenter, 1987; Just & Carpenter, 1992). (The ACT comprehension model (Anderson et al., 1977) is another good example of a model based on a production system architecture, but the READER models are better developed with respect to NL processing). There are many strong similarities. The READER models are built upon a general cognitive architecture (CAPS), just as NL-Soar is based on Soar. Both Soar and CAPS have productions as a fundamental component, and as a result both NL-Soar and READER embody the immediacy hypothesis. Both NL-Soar and READER are functionally complete models, in that they posit processes to handle multiple levels of comprehension (syntactic, semantic, referential). Both theories also model certain aspects of the time course of comprehension.

Though a full comparison of Soar and CAPS is beyond the scope of the present discussion, it is worth noting a number of key differences. CAPS is activation-based, and therefore deals with continuously-varying production strengths and memory traces, while the match in Soar is discrete. Soar posits a level of control built on top of the production system (problem spaces) and therefore introduces an automatic/deliberate distinction. The total activation in CAPS may be adjusted as a reflection of working memory capacity, while Soar has essentially no structural parametric variation. Soar has a continuous learning mechanism, which is absent in CAPS.

As a result of the underlying architectures and other assumptions of the models, there are several significant differences between NL-soar and the READER models. NL-Soar makes precise structure-sensitive predictions of garden path effects and parsing breakdown; presently it is not clear how READER could be applied in detail to the collection of constructions (including the cross-linguistic examples) that formed an important core of NL-Soar's predictions. On the other hand, NL-Soar cannot yet model performance variations due to individual differences in working memory capacity. Both NL-Soar and READER should be able to model individual differences based on knowledge (or skill) differences, by positing different sets of productions or associations. But READER does not provide the mechanisms (learning) by which these differences might arise as a function of experience. Although both models model the time course of comprehension, NL-Soar cannot model the critical differences in fixation times due to word frequency and length because it does not have a detailed model of lexical encoding and access. However, NL-Soar is able to make zero-parameter predictions of comprehension rate, due to Soar's temporal grounding.

9.4.2 Pritchett's model

As far as garden path theories are concerned, NL-Soar bears the closest resemblance to Pritchett's On-line Locality Constraint (OLLC) (Pritchett, 1992). Both models embody the Structural Garden Path Hypothesis (54), and as a result both exhibit extreme sensitivity to differences in syntactic structure. Indeed, Pritchett's original theory (Pritchett, 1988) provided the inspiration for a structural repair mechanism. Furthermore, both models are purely head-driven, bottom-up parsers.

There are several similarities and differences between the OLLC and NL-Soar's repair mechanism. The OLLC is essentially an abstract characterization of precisely what structural differences between the pursued and correct interpretations will yield garden path effects. As such, it represents the Structural Garden Path Hypothesis in its purest form: there is no commitment to particular computational processes, or even to a single path/reanalysis model. The NL-Soar theory, on the other hand, posits an explicit set of functional mechanisms to handle unproblematic ambiguities. Ultimately, even if a characterization such as the OLLC proves correct, there must be some account given of how the computational processes of comprehension yield such a characterization. Of course, NL-Soar predicts that an account such as the OLLC must ultimately be right—the repair mechanism in NL-Soar fails or succeeds purely as a function of the differences between the pursued and required syntactic structures.

Given these similarities, it is interesting to consider the status of the OLLC as a *grammatically-derived* theory. If structure-sensitivity is all that is required, NL-Soar is just as grammatically-based as the OLLC. But by grammatically-derived, Pritchett (1992) means more than this: the crucial fact is that the OLLC is formulated in terms of fundamental relations (dominance and government) of the grammar. The significance of this formulation is unclear, however—the particular form of the OLLC (a disjunction of dominance and government) has no independent grammatical status, and it is perhaps somewhat odd to expect any grammatical significance to obtain for relations computed *across different structures*, as is the case with the OLLC. On the other hand, we should expect, if the Structural Garden Path Hypothesis is correct, that *some* formulation of a reanalysis constraint should be possible in terms of grammatical primitives, since it is precisely such primitives that provide the language for talking about aspects of syntactic structure. Thus, it is not surprising that the locality constraint built into the generator for NL-Soar's snip operator can be readily formulated in terms of a simple grammatical construct—namely, the maximal projection[1].

[1] Under this view, the locality constraint of the snip operator is perhaps even simpler than the OLLC. A further advantage of snip is that this single mechanism covers all the necessary repairs, while an implementa-

The head-driven assumption of Pritchett's model was adopted in NL-Soar because it yields the minimal set of operators to construct the utterance model: no additional processes are required to build expectation structures corresponding to top-down predictions, and no processes are required to match incoming structure with predicted structure (see (Lehman et al., 1991a) for a description of an earlier version of NL-Soar that did have an expectation-based component).

While NL-Soar incorporates head-driven processing and embodies the structure-sensitivity of the OLLC, it differs significantly from the Generalized Theta Attachment (GTA) principle which governs ambiguity resolution in Pritchett's model. As noted in Chapter 2, the most serious problem with the GTA and any other syntactically-driven model of ambiguity resolution is the inability to account for interactive effects that have been established across a range of contexts, structural ambiguities, and experimental paradigms. Nevertheless, NL-Soar does share some predictions for attachment preferences with GTA, in particular, the preference for objects over subjects (§4.1.3).

9.4.3 Gibson's model

Gibson's 1991 model was the first to make detailed predictions across a broad range of both garden path and parsing breakdown phenomena, and is still the only theory apart from NL-Soar to do so.

Like NL-Soar and the On-line Locality Constraint, Gibson's model is a structure-sensitive theory. It incorporates a structural metric that assigns values to alternative interpretations. Garden path effects are predicted when the metric assigned to two interpretations differs by more than a constant factor, leading to the disposal of one of the interpretations. Thus, the Gibson model also embodies the Structural Garden Path hypothesis.

Though Gibson presents the theory as a memory overload model, it is unclear how to interpret the theory in mechanistic terms. Of course, a computational implementation of the theory can be constructed (and Gibson did construct one) that obtains efficiency gains by using the structural metric to prune the parsing search space. But such an implementation cannot be taken as a cognitive model in the strongly equivalent sense of Pylyshyn (1984), because the processing steps in the implementation are not held to correspond to the processing steps of the human comprehender. (The implementation applies the theory straightforwardly: it generates the alternative interpretations, computes the metric, and discards structures according to the pruning rule.) The architectural status of the various weights remains unexplicated. Thus, the present theoretical value of the metric is not to be found in its realization in a process model, but instead in its precise articulation of the structural differences that lead to garden path effects. This abstract interpretation of the theory places it in the same general class of theories as the OLLC, though the OLLC accomplishes the function somewhat more transparently.

tion of the OLLC may require both a structural reanalysis mechanism similar to snip, as well as a mechanism to handle the node-feature respecification necessary to repair some kinds of lexical ambiguities (Pritchett, 1992).

NL-Soar's explanation of processing breakdown is similar to the explanation provided by Gibson's theory, in that both identify the problem as primarily one of buffering uninterpreted NPs. Indeed, Gibson's analyses paved the way for the account presented here. Again, however, the structural metric must eventually be given a processing interpretation. NL-Soar's A/R set provides such a mechanistic theory (though not one that directly realizes Gibson's metric—the two theories do in fact make different predictions). Furthermore, the A/R set has some independent functional and computational motivation—and perhaps psychological motivation, as discussed above (§9.3.1).

9.5 Challenging areas for future work

NL-Soar not only raises many challenging issues to resolve in the current model, but potentially opens up new areas of theoretical and empirical inquiry. This section discusses just a few of these issues and areas.

First, of course, are the several empirical problems the model encounters on the present corpora. A number of missed predictions point to a possible problem with the way NL-Soar handles complements vs. adjuncts. The model also appears to somewhat overpredict acceptability of difficult embeddings. Here, the challenge will not just involve modifying the theory, but acquiring more reliable data as well.

NL-Soar is one of the few psycholinguistic models of sentence processing to incorporate continuous learning (beyond acquisition) as a central component. This feature, along with the automatic/deliberate distinction inherent in NL-Soar, may provide a way to unify the increasing amount of psychological data addressing this distinction. The deliberate garden path recovery model presented in Chapter 4 is just one example of how the theory can be applied. NL-Soar could, in general, open up new areas of study concerning the impact of learning on various aspects of parsing and interpretation.

Because NL-Soar is embedded in a general cognitive theory, it offers the opportunity to study the integration of language comprehension with other tasks, including language *generation*. In artificial intelligence, the integration of comprehension is already under way with Huffman's (1993) work on instructable agents. Other work in the NL-Soar project at CMU is investigating the low-level interleaving of comprehension and generation with other task operators in the demanding real-time environment of the NASA test director (the individual responsible for launching the space shuttle) (Nelson et al., 1994). Apart from the real-time integration of comprehension and generation processes, the generation work permits the exploration of shared knowledge sources and working memory structures.

Finally, the very simple structure of the A/R set makes it a good candidate for exploring neural foundations and developing network implementations. Stromswold, Alpert, Rausch, and Caplan (1993) have already carried out imaging experiments (PET) with subjects reading embedded clauses, identifying a potential locus for syntactic buffering within Broca's area.

9.6 Contributions and conclusion

This thesis began by setting forth four criteria for the comprehension model to be developed: breadth, depth, architectural basis, and functionality. NL-Soar satisfies all four criteria. It covers a broad range of sentence processing phenomena: garden path and unproblematic ambiguities, difficult and acceptable embeddings, modular and interactive ambiguity resolution effects, immediacy of interpretation, and the time course of comprehension. It accounts for the phenomena in depth: the theory makes successful predictions on a collection of over 100 items, including cross-linguistic constructions. The theory has an architectural basis: the Soar cognitive architecture, which provides the control structure, memory structures, and learning mechanism. Finally, the theory is functional: the model posits computational mechanisms that realize the functions of comprehension at multiple levels, and the model functions as a working comprehension system.

The architectural grounding proved to be theoretically and empirically fecund, contributing to a number of firsts for the model: for example, the first zero-parameter predictions of comprehension rate, the first detailed model of deliberate recovery from garden paths, and the first model of how learning might modulate modularity.

In short, NL-Soar provides new understanding of how human language comprehension can be immediate and real-time, yet extremely flexible; how it appears to effortlessly handle local ambiguities and embeddings most of the time, yet fail in certain situations; how it can be special-purpose and finely tuned to the task, yet tightly integrated with the rest of cognition; and how it can all be assembled from basic computational mechanisms that are fundamental to explaining many other aspects of intelligent behavior.

Bibliography

Abney, S. P. (1986). Licensing and parsing. In *Proceedings of NELS XVII*.

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.

Allen, J. (1987). *Natural Language Understanding*. Benjamin/Cummings, Menlo Park, CA.

Altmann, G. (1987). Modularity and interaction in sentence processing. In Garfield, J. L., editor, *Modularity in Knowledge Representation and Natural-Language Understanding*. MIT Press, Cambridge, MA.

Altmann, G. (1988). Ambiguity, parsing strategies, and computational models. *Language and Cognitive Processes*, 3:73–97.

Altmann, G., Garnham, A., & Dennis, Y. (1992). Avoiding the garden path: Eye movements in context. *Journal of Memory and Language*, 31:685–712.

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

Anderson, J., Kline, P., & Lewis, C. (1977). A production system model of language processing. In Just, M. A. & Carpenter, P. A., editors, *Cognitive Processes in Comprehension*. Lawrence Erlbaum, Hillsdale, NJ.

Anderson, J. R. (1973). Memory for information about individuals. *Memory and Cognition*, 5:430–442.

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

Anderson, J. R. & Bower, G. H. (1973). *Human Associative Memory*. Winston, Washington, D. C.

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.

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

Bever, T. G., Garrett, M. F., & Hurtig, R. (1973). The interaction of perceptual processes and ambiguous sentences. *Memory and Cognition*, 1:277–286.

Bickerton, D. (1992). Unified cognitive theory: You can't get there from here. *Behavioral and Brain Sciences*, 15(3):437–438.

Blank, G. D. (1989). A finite and real-time processor for natural language. *Communications of the ACM*, 32(10):1174–1189.

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.

Bower, G. H. & Morrow, D. G. (1990). Mental models in narrative comprehension. *Science*, 247:44–48.

Bransford, J. D., Barclay, J. R., & Franks, J. J. (1972). Sentence memory: a constructive versus interpretive approach. *Cognitive Psychology*, 3:193–209.

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

Carbonell, J. G., Knoblock, C. A., & Minton, S. (1989). PRODIGY: An integrated architecture for planning and learning. In VanLehn, K., editor, *Architectures for Intelligence*. Erlbaum, Hillsdale, NJ.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum, Hillsdale, NJ.

Cardie, C. & Lehnert, W. (1991). A cognitively plausible approach to understanding complex syntax. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 117–124.

Carpenter, P. A. & Daneman, M. (1981). Lexical retrieval and error recovery in reading: a model based on eye fixations. *Journal of Verbal Learning and Verbal Behavior*, 20:137–160.

Carpenter, P. A. & Just, M. A. (1977). Reading comprehension as eyes see it. In Just, M. A. & Carpenter, P. A., editors, *Cognitive Processes in Comprehension*. Lawrence Erlbaum, Hillsdale, NJ.

Carpenter, P. A. & Just, M. A. (1983). What your eyes do while your mind is reading. In Rayner, K., editor, *Eye Movements in Reading: Perceptual and Language Processes*. Academic Press, New York.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological Review*, 97(3):404–431.

Chomksy, N. (1964). On certain formal properties of grammars. *Information and Control*, 2:137–167.

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

Chomsky, N. (1973). Conditions on transformations. In Anderson, S. R. & Kiparsky, P., editors, *A Festschrift for Morris Halle*. Holt, Rinehart, and Winston, New York.

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

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

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

Church, K. W. (1980). On memory limitations in natural language processing. Technical Report MIT/LCL/TR-245, Laboratory for Computer Science, Massachusetts Institute of Technology.

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

Clifton, C., J. & Ferreira, F. (1989). Ambiguity in context. *Language and Cognitive Processes*, 4:SI 77–103.

Conati, C. & Lehman, J. F. (1993). Toward a model of student education in microworlds. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*.

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

Cowper, E. A. (1992). *A Concise Introduction to Syntactic Theory*. University of Chicago Press, Chicago.

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

Crocker, M. (1990). Principle-based sentence processing: A cross-linguistic account. Technical Report Research Paper 1, Human Communication Research Center, University of Edinburgh.

De Roeck, A., Johnson, R., King, M., Rosner, M., Sampson, G., & Varile, N. (1982). A myth about center-embedding. *Lingua*, 58:327–340.

Dell, G. S., McKoon, G., & Ratcliffe, R. (1983). The activation of antecedent information during the processing of anaphoric reference in reading. *Journal of Verbal Learning and Verbal Behavior*, 22:121–132.

Doorenbos, R. B. (1993). Matching 100,000 learned rules. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 290–296.

Duffy, S. A. & Rayner, K. (1990). Eye movements and anaphor resolution: Effects of antecedent typicality and distance. *Language and Speech*, 33(2):103–119.

Eady, J. & Fodor, J. D. (1981). Is center embedding a source of processing difficulty? Presented at the Linguistic Society of America Annual Meeting.

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

Ebbinghaus, H. D., Flum, J., & Thomas, W. (1984). *Mathematical Logic*. Springer-Verlag, New York.

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.

Ferreira, F. & Henderson, J. M. (1991). Recovery from misanalyses of garden-path sentences. *Journal of Memory and Language*, 30:725–745.

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

Fodor, J. A., Bever, T., & Garrett, M. (1974). *The Psychology of Language*. McGraw Hill, New York.

Fodor, J. A. & Garrett, M. (1967). Some syntactic determinants of sentential complexity. *Perception and Psychophysics*, 2(7):289–296.

Fodor, J. D. (1988). On modularity in syntactic processing. *Journal of Psycholinguistic Research*, 17(2):125–168.

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

Ford, M. (1983). A method for obtaining measures of local parsing complexity throughout sentences. *Journal of Verbal Learning and Verbal Behavior*, 22:203–218.

Ford, M., Bresnan, J., & Kaplan, R. M. (1982). A competence-based theory of syntactic closure. In Bresnan, J., editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA.

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.

Foss, D. J. & Jenkins, C. M. (1973). Some effects of context on the comprehension of ambiguous sentences. *Journal of Verbal Learning and Verbal Behavior*, 12:577–589.

Frank, R. E. (1992). *Syntactic Locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing Perspectives*. PhD thesis, University of Pennsylvania.

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

Frazier, L. (1985). Syntactic complexity. In Dowty, D. R., Karttunen, L., & Zwicky, A. M., editors, *Natural Language Parsing*. Cambridge University Press, New York.

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. & Fodor, J. D. (1978). The Sausage Machine: a new two-stage parsing model. *Cognition*, 6:291–325.

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.

Frazier, L. & Rayner, K. (1987). Resolution of syntactic category ambiguities: Eye movements in parsing lexically ambiguous sentences. *Journal of Memory and Language*, 26:505–526.

Frege, G. (1892). Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50. Translated as "On Sense and Reference", in Beach, P. T. and Black, M., *Translations from the Philosophical writings of Gottlob Frege*, Blackwell, Oxford, 1960.

Garnham, A. (1987). *Mental Models as Representations of Discourse and Text*. Ellis Horwood Limited, Chichester, West Sussex, England.

Gibson, E. & Pearlmutter, N. (1993). A corpus-based analysis of constraints on PP attachment to NPs. Manuscript, Carnegie Mellon University, Pittsburgh, PA.

Gibson, E., Pearlmutter, N., E., C.-G., & Hickok, G. (1993). Cross-linguistic attachment preferences: Evidence from english and spanish. In *6th Annual CUNY Sentence Processing Conference*, Amherst, MA.

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

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.

Gorrell, P. (1993). Incremental structure building and the determinism hypothesis. In *6th Annual CUNY Sentence Processing Conference*, Amherst, MA.

Green, G. (1976). Main clause phenomena in subordinate clauses. *Language*, 52:382–397.

Greene, S. B., McKoon, G., & Ratcliff, R. (1992). Pronoun resolution and discourse models. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18(2):266–283.

Grosz, B. J. & Sidner, C. (1986). Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3).

Hakes, D. T. (1972). Effects of reducing complement constructions on sentence comprehension. *Journal of Verbal Learning and Verbal Behavior*, 11:??–??

Hakes, D. T. & Foss, D. J. (1970). Decision processes during sentence comprehension: Effects of surface structure reconsidered. *Perception and Psychophysics*, 8(6):413–416.

Hickok, G. (1993). Parallel parsing: Evidence from reactivation in garden-path sentences. *Journal of Psycholinguistic Research*, 22(2):239–250.

Hindle, D. & Rooth, M. (1991). Structural ambiguity and lexical relations. In *Proceedings of the Association for Computational Linguistics*, pages 229–236.

Hobbs, J. & Bear, J. (1990). Two principles of parse preference. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, volume 3, pages 162–167.

Holmes, V. M. & O'Regan, J. K. (1981). Eye fixation patterns during the reading of relative clause sentences. *Journal of Verbal Learning and Verbal Behavior*, 20:417–430.

Hooper, J. & Thompson, S. (1973). On the applicability of root transformations. *Linguistic Inquiry*, 4:465–497.

Huffman, S. B. (1993). *Instructable Autonomous Agents*. PhD thesis, The University of Michigan. Department of Electrical Engineering and Computer Science. Forthcoming.

John, B. E. (1988). *Contributions to engineering models of human-computer interaction*. PhD thesis, Carnegie Mellon University.

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

Johnson-Laird, P. N. (1988). Reasoning by rule or model? In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*.

Johnson-Laird, P. N. & Byrne, R. M. J. (1991). *Deduction*. Lawrence Erlbaum, Hillsdale, NJ.

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

Just, M. A. & 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.

Katz, J. J. & Fodor, J. (1963). The structure of a semantic theory. *Language*, 39(2):170–210.

Kempen, G. & Vosse, T. (1989). Incremental syntactic tree formation in human sentence processing: a cognitive architecture based on activation decay and simulated annealing. *Connection Science*, 1(3):273–290.

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

Kimball, J. (1975). Predictive analysis and over-the-top parsing. In Kimball, J., editor, *Syntax and Semantics*, volume 4. Academic Press, New York.

Kintsch, W. & van Dijk T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, 85:363–394.

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

Kosy, D. W. (1986). Parsing conjunctions deterministically. In *Proceedings of the 24th ACL Conference*, pages 78–84, New York.

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

Lackner, J. R. & Garrett, M. F. (1972). Resolving ambiguity: Effects of biasing context in the unattended ear. *Cognition*, 1:359–372.

Laird (1991). Special section on integrated cognitive architectures. *SIGART Bulletin*, 2(4):12–184. Papers presented at the 1991 AAAI Spring Symposium on Integrated Intelligent Architectures.

Laird, J. E. (1988). Recovery from incorrect knowledge in Soar. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 618–623, AAAI.

Laird, J. E., Congdon, C. B., Altmann, E., & Doorenbos, R. (1993). *Soar User's Manual: Version 6 (Edition 1)*. Electrical Engineering and Computer Science Department, University of Michigan.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64.

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

Lehman, J. F., Lewis, R. L., & Newell, A. (1991a). 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.

Lehman, J. F., Lewis, R. L., & Newell, A. (1991b). Natural language comprehension in Soar: Spring 1991. Technical Report CMU-CS-91-117, School of Computer Science, Carnegie Mellon University.

Lehman, J. F., Lewis, R. L., & Newell, A. (1993a). Architectural influences on language comprehension. In Pylyshyn, Z., editor, *Cognitive Architecture*. (Publisher under selection). To appear.

Lehman, J. F., Newell, A., Polk, T. A., & Lewis, R. L. (1993b). The role of language in cognition: A computational inquiry. In Harman, G., editor, *Conceptions of the Human Mind*. Erlbaum, Hillsdale, NJ. In press.

Lewis, D. (1972). General semantics. In Davidson, D. & Harman, G., editors, *Semantics for Natural Language*. D. Reidel, Dordrecht, The Netherlands. Reprinted in Partee, B. H., ed., *Montague Grammar*, New York: Academic Press, 1976.

Lewis, H. R. & Papadimitriou, C. H. (1981). *Elements of the Theory of Computation*. Prentice-Hall, Englewood Cliffs, NJ.

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. (1993a). An architecturally-based theory of sentence comprehension. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*.

Lewis, R. L. (1993b). 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. To appear.

Lewis, R. L., Huffman, S. B., John, B. E., Laird, J. E., Lehman, J. F., Newell, A., Rosenbloom, P. S., Simon, T., & Tessler, S. G. (1990). Soar as a unified theory of cognition: Spring 1990. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pages 1035–1042. 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., Newell, A., & Polk, T. A. (1989). Toward a Soar theory of taking instructions for immediate reasoning tasks. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pages 514–521. Also in P. S. Rosenbloom, J. E. Laird, and A. Newell, eds., *The Soar Papers: Research on Integrated Intelligence*, MIT Press, Cambridge, MA, 1993.

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

MacDonald, M. C., Just, M. A., & Carpenter, P. A. (1993). Working memory constraints on the processing of syntactic ambiguity. *Cognitive Psychology*, 24:59–98.

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

Mani, K. & Johnson-Laird, P. N. (1982). The mental representation of spatial descriptions. *Memory and Cognition*, 10:181–18.

Marcus, M., Hindle, D., & Fleck, M. (1983). D-theory: Talking about talking about trees. In *Proceedings of the Association for Computational Linguistics*, volume 21.

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

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

Marslen-Wilson, W. (1973). Linguistic structure and speech shadowing at very short latencies. *Nature*, 244:522–523.

Marslen-Wilson, W. & Tyler, L. K. (1987). Against modularity. In Garfield, J. L., editor, *Modularity in Knowledge Representation and Natural-Language Understanding*. MIT Press, Cambridge, MA.

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

Mazuka, R., Itoh, K., Kiritani, S., Niwa, S., Ikejiru, K., & Naito, 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. (1988). *The Syntactic Phenomena of English, Volume 1*. The University of Chicago Press, Chicago.

McKoon, G. & Ratcliff, R. (1992). Inference during reading. *Psychological Review*, 99(3):440–466.

Mel'ĉuk, I. A. (1988). *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany, NY.

Miller (1990). Five papers on WordNet. *International Journal of Lexicography*, 3:235–312.

Miller, C. (1993). *Modeling Concept Acquisition in the Context of a Unified Theory of Cognition*. PhD thesis, The University of Michigan. Department of Electrical Engineering and Computer Science.

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., R., B. 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.

Mitchell, D. (1987). Lexical guidance in human parsing: Locus and processing characteristics. In Coltheart, M., editor, *Attention and Performance*, volume 12. 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.

Nelson, G., Lehman, J. F., & John, B. E. (1994). Experiences in interruptible language processing. To appear in AAAI Spring Symposium on Active Natural Language Processing.

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

Newell, A. (1973b). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In Chase, W. G., editor, *Visual Information Processing*. Academic Press, New York.

Newell, A. (1980). 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. (1982). The knowledge level. *Artificial Intelligence*, 18:87–127. (Also in *AI Magazine*, *Vol 2,* Summer 1981, pp. 1-20; and as CMU CSD Technical Report, Aug. 1981).

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

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

Newell, A., Yost, G., Laird, J. E., Rosenbloom, P. S., & Altmann, E. (1991). Formulating the problem-space computational model. In Rashid, R. F., editor, *CMU Computer Science: A 25th Anniversary Commemorative*. Addison-Wesley, Reading, MA.

Nicol, J. L. & Pickering, M. J. (1993). Processing syntactically ambiguous sentences: Evidence from semantic priming. *Journal of Psycholinguistic Research*, 22(2):207–237.

Oakhill, J., Garnham, A., & Vonk, W. (1989). The on-line construction of discourse models. *Language and Cognitive Processes,* 4:SI 263–286.

Pearlmutter, N. J. & MacDonald, M. C. (1992). Plausibility and syntactic ambiguity resolution. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 498–503, Hillsdale, NJ. Lawrence Erlbaum.

Pickering, M. & Barry, G. (1991). Sentence processing without empty categories. *Language and Cognitive Processes*, 6:229–259.

Polk, T. A. (1992). *Verbal Reasoning*. PhD thesis, School of Computer Science, Carnegie Mellon University. Also available as Computer Science tech report CMU-CS-92-178.

Polk, T. A. & Newell, A. (1988). Modeling human syllogistic reasoning in Soar. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, pages 181–187. Also in P. S. Rosenbloom, J. E. Laird, and A. Newell, eds., *The Soar Papers: Research on Integrated Intelligence*, MIT Press, Cambridge, MA, 1993.

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

Pritchett, B. L. (1991). Head position and parsing ambiguity. *Journal of Psycholinguistic Research*, 20:251–270.

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

Pustejovsky, J. & Boguraev, B. (1993). Lexical knowledge representation and natural language processing. *Artificial Intelligence*, 63:193–223.

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

Rayner, K., Carlson, M., & Frazier, L. (1983). The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior*, 22:358–374.

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

Rochemont, M. S. & Culicover, P. W. (1990). *English Focus Constructions and the Theory of Grammar*. Cambridge University Press, Cambridge, England.

Rosenbaum, P. (1967). *The Grammar of English Predicate Complement Constructions*. MIT Press, Cambridge, MA.

Rosenbloom, P. S., Laird, J. E., & Newell, A., editors (1993a). *The Soar Papers: Research on Integrated Intelligence*. MIT Press, Cambridge, MA.

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

Rosenbloom, P. S., Newell, A., & Laird, J. E. (1991). Towards the knowledge level in Soar: The role of the architecture in the use of knowledge. In VanLehn, K., editor, *Architectures for Intelligence*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Sanford, A. J. & Garrod, S. C. (1989). What, when, and how?: Questions of immediacy in anaphoric reference resolution. *Language and Cognitive Processes*, 4:SI 235–262.

Schank, R. C. & Riesbeck, C. K. e. (1981). *Inside Computer Understanding*. Lawrence E. Erlbaum, Hillsdale, NJ.

Schlesinger, I. M. (1968). *Sentence Structure and the Reading Process*. Mouton, The Hague.

Schmalhofer, F. & Glavanov, D. (1986). Three components of understanding a programmer's manual: Verbatim, propositional, and situational representations. *Journal of Memory and Language*, 25:279–294.

Simon, H. A. & Hayes, J. (1979). Understanding written problem instructions. In Simon, H. A., editor, *Models of Thought: Volume 1*. Yale University Press, New Haven, CT.

Simon, H. A. & Zhang, G. (1985). Stm capacity for chinese words and idioms: Chunking and the acoustical loop hypotheses. *Memory and Cognition*, 13:193–201.

St. John, M. F. & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, 46:217–257.

Steedman, M. & Altmann, G. (1989). Ambiguity in context: A reply. *Language and Cognitive Processes*, 4:SI 105–122.

Steier, D. M., Laird, J. E., Newell, A., Rosenbloom, P. S., Flynn, R. A., Golding, A., Polk, T. A., Shivers, O. G., Unruh, A., & Yost, G. R. (1987). Varieties of learning in Soar: 1987. In *Proceedings of the Fourth International Workshop on Machine Learning*, pages 300–311.

Steier, D. M., Lewis, R. L., Lehman, J. F., & Zacherl, A. L. (1993). Combining multiple sources of knowledge in an integrated intelligent system. *IEEE Expert*, 8(3):35–44.

Stolz, W. S. (1967). A study of the ability to decode grammatically novel sentences. *Journal of Verbal Learning and Verbal Behavior*, 6:867–873.

Stowe, L. (1986). Evidence for on-line gap location. *Language and Cognitive Processes*, 1:277–245.

Stromswold, K., Alpert, N., Rausch, S., & Caplan, D. (1993). The neural modularity of sentence processing. Presented at the Sixth Annual CUNY Sentence Processing Conference, Amherst, MA.

Swinney, D. (1979). Lexical access during sentence comprehension: (Re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, 18:645–659.

Swinney, D. A. & Osterhout, L. (1990). Inference generation during comprehension. In Graesser, A. C. & Bower, G. H., editors, *Inferences and Text Comprehension*. Academic Press, San Diego.

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

Tambe, M. & Rosenbloom, P. S. (1993). On the masking effect. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 526–533.

Tanenhaus, M. K. & Carlson, G. (1989). Lexical structure and language comprehension. In Marslen-Wilson, W., editor, *Lexical Representation and Process*. MIT Press, Cambridge, MA.

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.

Tanenhaus, M. K., Garnsey, S. M., & Boland, J. (1990). Combinatory lexical information and language comprehension. In Altmann, G. T. M., editor, *Cognitive Models of Speech Processing*. MIT Press, Cambridge, MA.

Taraban, R. & McClelland, J. L. (1988). Constituent attachment and thematic role assignment in sentence processing: Influences of content based expectations. *Journal of Memory and Language*, 27:597–632.

Taraban, R. & McClelland, J. L. (1990). Parsing and comprehension: A multiple-constraint view. In Balota, D. A., Flores d'Arcais, G. B., & Rayner, K., editors, *Comprehension processes in reading*. Lawrence Erlbaum, Hillsdale, NJ.

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

Trueswell, J. C. & Tanenhaus, M. K. (1992). Consulting temporal context during sentence comprehension: Evidence from the monitoring of eye movements in reading. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 492–497, Hillsdale, NJ. Lawrence Erlbaum.

Tyler, L. K. (1989). The role of lexical representations in language comprehension. In Marslen-Wilson, W., editor, *Lexical Representation and Process*. MIT Press, Cambridge, MA.

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

Tyler, L. K. & Marslen-Wilson, W. D. (1982). Processing utterances in discourse contexts: On-line resolution of anaphors. *Journal of Semantics*, 1:297–315.

Van Dijk, T. A. & Kintsch, W. (1983). *Strategies of Discourse Comprehension*. Academic Press, Orlando, FL.

Vera, A. H., Lewis, R. L., & Lerch, F. J. (1993). Situated decision-making and recognition-based learning: Applying symbolic theories to interactive tasks. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*.

Wanner, E. (1980). The ATN and the Sausage Machine: Which one is baloney? *Cognition*, 8:209–225.

Wanner, E., Kaplan, R., & Shiner, S. (1975). Garden paths in relative clauses. Unpublished manuscript, Harvard University.

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

Weckerly, J. & Elman, J. L. (1992). A PDP approach to processing center-embedded sentences. In *Proceedings of the Twelfth Annual Meeting of the Cognitive Science Society*, pages 414–419.

Weinberg, A. (1991). A parsing theory for the Nineties: Minimal commitment. Unpublished manuscript, University of Maryland at College Park.

Weinberg, A. (1993). Parameters in the theory of sentence processing: Minimal commitment theory goes east. *Journal of Psycholinguistic Research*, 22(3):339–364.

Whittemore, G. & Ferrara, K. (1990). Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 23–30.

Wiesmeyer, M. D. (1992). *An Operator-based Model of Human Covert Visual Attention*. PhD thesis, The University of Michigan, Department of Electrical Engineering and Computer Science. Available as tech report CSE-TR-123-92.

Wilks, Y. (1975). An intelligent analyzer and understander of English. *Communications of the ACM*, 18(5):264–274.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology*, 3:1–191.

Winograd, T. (1983). *Language as a Cognitive Process, Volume 1: Syntax*. Addison Wesley, Reading, MA.

Yngve, V. H. (1960). A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society*, 104(5):444–466.

Young, R. M. (1993). Functionality matters: Capacity constraints and Soar. In Steier, D. & Mitchell, T., editors, *Mind Matters: Contributions to Cognitive and Computer Science in Honor of Allen Newell*. Erlbaum, Hillsdale, NJ. To appear.