An On-line Computational Model of Human Sentence Interpretation: A Theory of the Representation and Use of Linguistic Knowledge

Daniel Jurafsky

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAR 1992**	2. REPORT TYPE	3. DATES COVERED **00-00-1992 to 00-00-1992**
4. TITLE AND SUBTITLE **An On-Line Computational Model of Human Sentence Interpretation: A Theory of the Representation and Use of Linguistic Knowledge**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Berkeley,Department of Electrical Engineering and Computer Sciences,Berkeley,CA,94720**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

This dissertation presents a model of the human sentence interpretation process, which attempts to meet criteria of adequacy imposed by the different paradigms of sentence interpretation. These include the need to produce a high-level interpretation, to embed a linguistically motivated grammar, and to be compatible with psycholinguistic results on sentence processing. The model includes a theory of grammar called Construction-Based Interpretative Grammar (CIG) and an interpreter which uses the grammar to build an interpretation for single sentences. An implementation of the interpreter has been built called Sal. Sal is an on-line interpreter, reading words one at a time and updating a partial interpretation of the sentence after each constituent. This constituent-by-constituent interpretation is more fine-grained and hence more on-line than most previous models. Sal is strongly interactionist in using both bottom-up and top-down knowledge in an evidential manner to access a set of constructions to build interpretations. It uses a coherence-based selection mechanism to choose among these candidate interpretations, and allows temporary limited parallelism to handle local ambiguities. Sal's architecture is consistent with a large number of psycholinguistic results. The interpreter embodies a number of strong claims about sentence processing. One claim is uniformity, with respect to both representation and process. In the grammar, a single kind of knowledge structure, the grammatical construction, is used to represent lexical, syntactic, idiomatic, and semantic knowledge. CIG thus does not distinguish between the lexicon, the idiom dictionary, the syntactic rule base, and the semantic rule base. Uniformity in processing means that there is no distinction between the lexical analyzer, the parser, and the semantic interpreter. Because these kinds of knowledge are represented uniformly, they can be accessed, integrated, and disambiguated by a single mechanism. A second claim the interpreter embodies is that sentence processing is fundamentally knowledge-intensive and expectation-based. The representation and integration of constructions uses many diverse types of linguistic knowledge. Similarly, the access of constructions is sensitive to top-down and bottom-up, syntactic and semantic knowledge, and the selection of constructions is based on coherence with grammatical knowledge and the interpretation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**220**	

An On-line Computational Model of Human Sentence Interpretation:
A Theory of the Representation and Use of Linguistic Knowledge

An On-line Computational Model of Human Sentence Interpretation: A Theory of the Representation and Use of Linguistic Knowledge

by

Daniel Jurafsky

Abstract

This dissertation presents a model of the human sentence interpretation process, which attempts to meet criteria of adequacy imposed by the different paradigms of sentence interpretation. These include the need to produce a high-level interpretation, to embed a linguistically motivated grammar, and to be compatible with psycholinguistic results on sentence processing.

The model includes a theory of grammar called Construction-Based Interpretive Grammar (CIG) and an interpreter which uses the grammar to build an interpretation for single sentences. An implementation of the interpreter has been built called Sal.

Sal is an on-line interpreter, reading words one at a time and updating a partial interpretation of the sentence after each constituent. This *constituent-by-constituent* interpretation is more fine-grained and hence more on-line than most previous models. Sal is strongly interactionist in using both bottom-up and top-down knowledge in an evidential manner to access a set of constructions to build interpretations. It uses a coherence-based selection mechanism to choose among these candidate interpretations, and allows temporary limited parallelism to handle local ambiguities. Sal's architecture is consistent with a large number of psycholinguistic results.

The interpreter embodies a number of strong claims about sentence processing. One claim is *uniformity*, with respect to both representation and process. In the grammar, a single kind of knowledge structure, the *grammatical construction*, is used to represent lexical, syntactic, idiomatic, and semantic knowledge. CIG thus does not distinguish between the lexicon, the idiom dictionary, the syntactic rule base, and the semantic rule base. Uniformity in processing means that there is no distinction between the *lexical analyzer*, the *parser*, and the *semantic interpreter*. Because these kinds of knowledge are represented uniformly, they can be accessed, integrated, and disambiguated by a single mechanism.

A second claim the interpreter embodies is that sentence processing is fundamentally *knowledge-intensive* and expectation-based. The *representation* and *integration* of constructions uses many diverse types of linguistic knowledge. Similarly, the *access* of constructions is sensitive to top-down and bottom-up, syntactic and semantic knowledge, and the *selection* of constructions is based on coherence with grammatical knowledge and the interpretation.

Acknowledgements

This dissertation would not have been possible without the support and advice of many people. First, my advisor Robert Wilensky, who has given me guidance and generous support throughout my dissertation work. He carefully read and critiqued my work, and our discussions resulted in many of the ideas presented here. The other members of my committee, Charles Fillmore, who was also my undergraduate advisor, Jerry Feldman, and Paul Kay, were extremely supportive and helpful throughout.

I owe an especial debt of gratitude to Nigel Ward, Graeme Hirst, and Jane Edwards, who read every chapter, and whose comments improved the dissertation immeasurably, and to Peter Norvig, George Lakoff, Marti Hearst, Ted Gibson, Terry Regier, Paul Hilfinger, and Adele Goldberg, whose advice and criticism over the years it has been an honor to receive.

Many thanks to my officemates, in mahjong order: Michael Braverman, always ready with a parable or a proof at just the right moment, and Jim Martin and Jim Mayfield, who proved to me that competence and professionalism could be invested with a deep humanitarianism, and Dekai Wu for those early heady days at Sufficient Grounds, and Nigel Ward, for his inspiring combination of encyclopedic knowledge and a strong critical sense; they invested my years in Evans Hall with wisdom and humor, as did the rest of the Berkeley AI research group, with whom it has been a pleasure to be associated: David Chin, Anthony Albert, Marti Hearst, Narciso Jaramillo, Michael Schiff, Terry Regier, Jordan Zlatev, Grazyna Wojcieszko, Charles Cox, Eric Karlson, HuaSheng Su, and Marc Luria (in no particular order).

For financial support, I would like to thank the taxpayers for providing funding throughout much of my dissertation research, administered through Defense Advanced Research Projects Agency (DoD), monitored by the Space and Naval Warfare Systems Command under N00039-88-C-0292, and the Office of Naval Research, under contract N00014-89-J-3205.

I don't think I could have made it without Nervous for Nigel (thanks to Jim Mayfield, who began it all, and then starting with the bassists: Pearl Chow, Eric Enderton, Erin Dare, Terry Regier, and Dekai Wu), and I thank them for all the Sundays and for the crazy, cruel, twisted, burning, sweet, time we had, and thanks, Terry, for the hammers.

Many people made the Computer Science Department more than just an office, of course especially Sharon Tague, whose contributions mere words are insufficient to describe, and Liza Gabato and Regina Thomas and Kathryn Crabtree, and from the ground up Nina Amenta, Ronitt Rubinfeld, Diane Hernek, Dana Randall, Mark Sullivan, Seth Teller, Ramon Caceres, Ziv Gigus, Caroline Helmke, Sandy Irani, Stuart Sechrest, and Will Evans. Especial thanks to Grazyna Wojcieszko and to the Ecole Central Paris.

Out in the real world my life would have been dull and pathetic without the Haste Street Gang, in order of appearance: Doug and Lynn Marinaro, Mike Bauer, Lisa Lum, Anna Baldessari, David Kwan, Ellen Baird, Hilary Bays, Debby Quinn, Kevin Sayama, Erin Dare, Janice Sugiyama, Sonia Couprie, and Amie Wilkinson, Giovanni Vitiello, and Michael Singman; and naturally Bruce and Peggy Winkelman, and Robby Jurafsky and Barbara Drexler, and the Dim Sum Dozen, in alphabetic order: Cary, Doris, Elaine, Eric, Godfrey, Irene, Rich, Sam, Shirley, Susan, and of course Vincent for the Ninja Turtles drawings.

Finally, my deepest thanks goes to my parents, whose faith and support and love kept me going all the way.

Contents

Chapter 1

Introduction

> "To parse a sentence is to relate it to a general description of a language"
>
> — *Hays (1966)*

> "If our goal as understanders is to extract a meaning from its language-encoded form, then the question we must ask is this: What is the best possible process to decode natural language?"
>
> — *Riesbeck & Schank (1978)*

> "An adequate theory of language comprehension must do more than describe the means by which the individual sentences of a text are processed and integrated into a coherent structure representing the meaning of the entire text. It must identify the *principles* determining the analysis of the input. . . "
>
> — *Frazier (1987a)*

As seen in these passages, there is no shared paradigm of what constitutes the nature and significance of research in language understanding. One paradigm, which might be called the *linguistic paradigm*, expressed here by Hays, is concerned with the relation between a computational model and linguistic theories of language structure. The second, *computational*, paradigm tends to be interested in the computationally best process for computing the meaning or structure of a sentence. Finally, the *psychological paradigm*, expressed here by Frazier, concerns itself with psychological modeling of the temporal processes which accompany human interpretation of language, and the expression of general principles which determine this processing.

The goals and the domain of study expressed by each of these paradigms are frequently assumed to be incompatible. Thus although the sentence interpretation process has received a great amount of attention in the cognitive science community, most models have tended to address very limited subparts of the problem of interpreting an utterance. By focusing on subproblems, such as lexical access, or syntactic disambiguation, or efficient parsing, these models do not generalize well enough to deal with broader sentence-interpretation issues.

But there is no reason why a model of human sentence interpretation must limit itself to a single paradigm, particularly in this era of interdisciplinary studies and cognitive science. This dissertation proposes a model of the human sentence interpretation process which attempts to address the fundamental criteria imposed by each of these paradigms. The model consists of two components:

- A theory of grammar called *Construction-Based Interpretive Grammar* (CIG), part of a family of theories called *Construction Grammars* (Fillmore *et al.* 1988; Kay 1990; Lakoff 1987), which represents knowledge of language as a collection of uniform structures called *grammatical constructions* representing lexical, syntactic, semantic, and pragmatic information.
- A semantic interpreter named *Sal* (after the well-known Erie Canal mule of songdom), which includes:
  - a *working store* which allows multiple constructions and interpretations to be considered in parallel.
  - an evidential *access* function, which uses different knowledge sources to guide its search for the correct constructions to access in an interactionist manner.
  - an information-combining operation called *integration*, which augments a unification-like operation with knowledge about the semantic representation language and with a mechanism for functional application.
  - a *selection* algorithm which prefers interpretations which are more *coherent* and which prunes low-ranked interpretations.

This characterization of any theory of interpretation as including sub-theories of *access*, *integration*, and *selection* is a very general one, frequently applied to models of the lexicon, for example. These three sub-theories will be used to structure the dissertation; the architecture of the interpreter will be described by giving specific proposals for an *access function*, an *interpretation function*, and a *selection function*, and each is described by a chapter.

1.1 Criteria for a Theory of Interpretation

This is a particularly exciting time to study computational models of language processing. Recent years, and particularly the last decade, have produced a cornucopia of experimental results from psycholinguistics. Many, if not most, modern linguistics theories have begun to be seriously concerned with psychological and computational issues. And everywhere computational results and models abound. It is the beginnings of a convergence of interests of these fields which make the development of such a model possible.

But modeling sentence understanding is difficult as much because of what we know as because of what we don't know. The more each paradigm requires of a successful model, the more there is a temptation to avoid these requirements by building models which do not stray beyond the bounds of an individual field. To avoid these problems, this section proposes an interdisciplinary set of broad-ranging criteria of adequacy for a theory of human sentence interpretation. The first criterion of **Functional Adequacy** constrains the nature of the interpretation.

> **Functional Adequacy:** *An interpreter must produce a representation which is rich and complete enough to function as an interpretation of the sentence in a larger model of language understanding.*

The functional adequacy criterion is a definitional one for an interpreter which is intended to model human processing. It is the necessity of meeting this criterion which distinguishes an interpreter, which must meet semantic and functional constraints on its representation, from a parser, which need not meet semantic constraints, or a lexical model, which need not even meet syntactic constraints. Without such a criterion, it is too easy to build a model of language processing whose defects are hidden by assuming that some as-yet-undefined module will solve them.

A processing model which accounts only for lexical access, or for syntactic parsing, may fail when a solution to the more general problem of sentence interpretation is called for. For example many models of lexical access account quite successfully for psycholinguistic results on lexical access. However none of these models deal with syntactic access. Similarly, no models of syntactic rule access can model the psycholinguistic results on lexical access. Because most models treat only one of lexical or syntactic access, the incompatibility between the approaches is not apparent. If either model was extended to deal with the other problem, however, the incompatibility would become clear, and might suggest changes in either model. Treating the problem of accessing linguistic knowledge in this piecemeal way can be avoided by adopting the criterion of Functional Adequacy.

The second criterion for an interpreter is that of **Representational Adequacy:**

> **Representational Adequacy:** *An interpreter must include a declarative and linguistically motivated representation of linguistic knowledge.*

This criterion insures that the representational basis of the processing model meets independent linguistic criteria for linguistic knowledge, particularly the need to capture relevant linguistic generalizations and account for the creativity of the language faculty.

Meeting the criterion of representational adequacy also requires that the linguistic knowledge used by the interpreter include more than just phonological or syntactic information. In order to produce an interpretation of a sufficient richness and completeness, the interpreter must bring to bear a large and varied collection of semantic, pragmatic, and world knowledge.

The final criterion concerns psychological validity:

> **Psychological Adequacy:** *An interpreter must meet standards of psycholinguistic and general cognitive validity.*

The criterion of Psychological Adequacy requires that the theory account in a principled manner for psycholinguistic results. A number of such results will be discussed in Chapters 4–7; the following list summarizes some of these results and the chapters in which they are discussed and modeled:

- the on-line nature of the language interpretation process (see Chapter 6)
- the parallel nature and time course of lexical, idiomatic and syntactic access (see Chapter 5)
- the context-dependence of the access point (see Chapter 5)
- the use of frequency information in access and in selection (see Chapters 5 and 7)
- the use of lexical knowledge such as valence, subcategorization, and thematic roles in integration (see Chapter 6)
- the nature and time-course of gap-filling (see Chapter 6)
- the use of expectations in selection (see Chapter 7)

Previous models of human language interpretation have generally focused on individual parts of these three criteria. For example many processing models which are associated with linguistic theories, such as Ford *et al.* (1982) (LFG), Marcus (1980) (EST) or the Government and Binding parsers such as Johnson (1991) or Fong (1991) include no semantic knowledge. That is, these are all models of *parsing*, and hence do not meet the criterion of functional adequacy. Alternatively, some models such as Riesbeck & Schank (1978), Birnbaum & Selfridge (1981), DeJong (1982) and others of the Yale school, have emphasized semantic knowledge but ignored syntactic knowledge. These models fail to meet representational adequacy. Some models, such as Lytinen (1986), do address representational adequacy by representing both syntactic and semantic knowledge, but, like these other classes of models, fail to meet the criterion of psychological adequacy.

Many models which derive from the psycholinguistic community and hence concentrate on psychological adequacy suffer by limiting their scope to lexical access; this includes the *cohort* model of Marslen-Wilson, or the *logogen* model of Morton. Again, by building a model of lexical access which ignores larger structures (e.g., syntactic rules or grammatical constructions), these models meet neither functional nor representational adequacy.

Because most models of human language processing have thus focused on either syntactic parsing or lexical access, very few cognitive models of interpretation have been proposed. Some more complete models have been proposed (such as Hirst (1986), Kurtzman (1985), Kintsch (1988), and Riesbeck & Schank (1978)), and these will be examined in further depth in Chapter 4. Relevant sections of Chapters 4–7 will concentrate on other models in more detail.

1.2 Motivating the Model

The criterion of psychological adequacy requires that the model account in a principled manner for psycholinguistic results concerning sentence processing; in the last ten years many such results have become available. Sal is an idealized model, and as such there is not a detailed *quantitative* fit with data such as the exact millisecond timing of events, but Sal is *qualitatively* consistent with all of the results summarized in Figures 1.1–1.4.

Considering the criteria in §1.1 and the linguistic and psycholinguistic phenomena summarized in these figures leads us to a number of properties that must be true of an interpreter like Sal and an embedded grammar like CIG. Most of these properties follow from the criteria and evidence, while some draw also on Occam's razor.

Access		
Lexical constructions are accessed in parallel.	Swinney (1979) Tanenhaus *et al.* (1979) Tyler & Marslen-Wilson (1982)	§5.1
Idioms are accessed in parallel.	Cacciari & Tabossi (1988)	§5.1
Syntactic constructions are accessed in parallel.	Kurtzman (1985) Gorrell (1987) and (1989) MacDonald *et al.* (in press)	§5.1
More frequent constructions are accessed more easily.	Marslen-Wilson (1990) Tyler (1984) Zwitserlood (1989) Simpson & Burgess (1985) Salasoo & Pisoni (1985)	§5.1
The *access-point* of a construction is not immediate, and varies on the context and on the construction.	Swinney & Cutler (1979) Cacciari & Tabossi (1988)	§5.1.1

Figure 1.1: Psycholinguistic Data on Access

Selection		
Selection Pruning		
Prune when one interpretation has a much more specific expectation	*#The grappling hooks onto the enemy ship.*	§7.3
Prune when one interpretation has a much more frequent expectation	*#The old man the boats.*	§7.6.3
Prune when one interpretation is much more coherent than the other.	*#The horse raced past the barn fell.*	§7.3
Selection Preference		
Prefer arguments to adjuncts	Ford *et al.* (1982)	§7.6
Prefer to fill expected constituents (Extraposition vs Pronominal It)	*It frightened the child that John wanted to visit the lab.* Crain & Steedman (1985)	§7.6.4
Selection preferences make references to lexical, syntactic, and semantic knowledge	Taraban & McClelland (1988) Stowe (1989) Trueswell & Tanenhaus (1991) Pearlmutter & MacDonald (1991) Zwitserlood (1989)	§7.6

Figure 1.2: Linguistic and Psycholinguistic Data on Selection

Integration		
Building an interpretation is an on-line process, occurring in a *constituent-by-constituent* manner, without waiting for the end of a sentence or clause.	Swinney (1979) Tanenhaus *et al.* (1979) Tyler & Marslen-Wilson (1982) Marslen-Wilson *et al.* (1988) Potter & Faulconer (1979) Marslen-Wilson (1975)	§6.2.3
The processor uses lexical valence and control knowledge in integration, including semantic and thematic knowledge.	Mitchell & Holmes (1985) Shapiro *et al.* (1987) Clifton *et al.* (1984) Boland *et al.* (1990) Tanenhaus *et al.* (1989)	
The processor experiences difficulty when encountering "filled gaps" in non-subject position.	Crain & Fodor (1985) Tanenhaus *et al.* (1985) Stowe (1986) Garnsey *et al.* (1989) Tanenhaus *et al.* (1989) Kurtzman *et al.* (1991)	§6.5.3
The processor does not experience the filled-gap effect in subject gaps.	Stowe (1986)	§6.5.3
The processor integrates distant fillers directly into the predicate, rather than mediating through an empty category.	Pickering & Barry (1991) Boland & Tanenhaus (1991)	§6.5.2

Figure 1.3: Psycholinguistic Data on Integration

Representation		
Inflection is represented distinctly from derivation	Stanners *et al.* (1979) Cutler (1983)	§3.7.3
Semantic constraints on constituents	*How three are they?	§3.4.3

Figure 1.4: Linguistic and Psycholinguistic Data on Representation

The architecture of the interpreter Sal is based on four properties: it is *on-line*, *parallel*, *interactive*, and *uniform*. Each of these properties is described in detail in Chapter 4.

Sal is *on-line* because it maintains an interpretation for the utterance at all times; it updates the interpretation in a *constituent-by-constituent* manner, which is a much more fine-grained and on-line method than the *rule-to-rule* method Bach (1976) which is used by most other models. The first part of Figure 1.3 summarizes a number of psycholinguistic results which indicate that human sentence processing is on-line in this manner rather than producing an interpretation only after a complete syntactic interpretation for the sentence has been produced.

Sal is *parallel* because it can maintain more than one interpretation simultaneously, although only for a limited time. The use of parallelism is motivated by a number of psycholinguistic results summarized in Figure 1.1. Parallelism in *lexical* access is a feature of all modern lexical-access models, and a number of recent psycholinguistic results indicate that the interpreter keeps parallel *syntactic* and *idiomatic* representations as well. Maintaining parallel representations thus allows a uniform treatment of lexical, idiomatic, and syntactic processing.

Sal is *interactive* because it allows syntactic, semantic, and higher-level expectations to help access linguistic information, integrate constructions into the interpretation, and choose among candidate interpretations. The interactive nature of the architecture is motivated by psycholinguistic results on access, integration, and selection. Results on integration and selection are summarized in Figures 1.3 and 1.2, respectively. Results on access are more mixed, and are discussed in more detail in §5.5. Our position on interactionism is incompatible with a strong version of Fodor's (1983) Modularity Hypothesis, in which semantic and contextual knowledge is unable to affect lower-level linguistic processing. Sal's architecture might, however, be compatible with a weaker version, in which semantic and contextual knowledge could affect lower-level processing, but world knowledge and the general reasoning capacity could not.

Sal is *uniform* because a single interpretation mechanism accounts for the access, integration, and selection of structures at all levels of sentence processing. Thus there is no distinction between the *lexical analyzer*, the *parser*, and the *semantic interpreter* — Sal performs all these functions in a unified way. This uniformity is motivated partly by the psycholinguistic evidence on access summarized in Figure 1.1, and also by Occam's razor; performing each of these tasks with one mechanism is more efficient than proposing distinct ones. The uniformity of the architecture is possible because the CIG grammar is uniform as well. Words, idioms, syntactic structures, and semantic interpretation rules are all uniformly represented as *grammatical* constructions.

This brings us to a discussion of the grammar. The criteria and the linguistic evidence lead to four properties that hold of CIG, and which we claim should hold in general of grammatical theories which are embedded in models of interpretation. The grammar must be *motivated*, *declarative*, *information-rich*, and *uniform*.

CIG is *motivated* because it is subject to the linguistic requirements of accounting for creativity and of capturing generalizations. It is the criterion of Representational Adequacy which requires an interpreter's grammar to be motivated.

CIG is *declarative* because it is *non-derivational*, or *non-constructive*. The grammar does not include the *derivational algorithm* which is a part of derivational grammars. Motivation for a declarative grammar is discussed in detail in Chapter 2.

CIG is *information-rich* because it includes information from various domains of linguistic knowledge, including phonological, syntactic, semantic, pragmatic, and frequency information.

Arguments for information-rich grammar are discussed in Chapter 3.

CIG is *uniform* because lexical entries, idioms, and syntactic structures are all represented uniformly as grammatical constructions. As was true with Sal, this uniformity is motivated partly by the psycholinguistic evidence on access summarized in Figure 1.1, and also by Occam's razor.

1.3 An Overview of the Model

Having discussed the motivation for the grammar and the interpreter, this section proceeds to sketch the architecture of the model itself. This section concentrates on describing and motivating the four sub-theories of the model: *representation*, *access*, *integration*, and *selection*. The following sections present a trace of the interpretation of a simple sentence and an outline of the rest of the dissertation.

1.3.1 The Grammar

The grammar that is embedded in Sal is an implementation of a linguistic theory called *Construction-Based Interpretive Grammar* (CIG), part of a family of theories called *Construction Grammars* (Fillmore *et al.* 1988; Kay 1990; Lakoff 1987). CIG defines a grammar as a declarative collection of structures called *grammatical constructions* which resemble the constructions of traditional pre-generative grammar. Each of these constructions represent information from various domains of linguistic knowledge, including phonological, syntactic, semantic, pragmatic, and frequency information. Thus the grammar constitutes a database of these constructions, which might be called a *"constructicon"* (on the model of the word *lexicon*). Allowing a construction to include semantic and pragmatic knowledge as well as syntactic knowledge helps CIG to meet the constraint of Functional Adequacy.

Lexical entries, idioms, and syntactic structures are all represented uniformly as grammatical constructions. Thus the "constructicon" subsumes the lexicon, the syntactic rule base, and the idiom dictionary assumed by other theories. Using a single representation for linguistic knowledge allows a very general mechanism for language understanding — lexical access, idiom processing, syntactic parsing, and semantic interpretation are all done by the same mechanism using the same knowledge base.

Like many recent theories of grammar (such as Pollard & Sag 1987; Bresnan 1982a; Uszkoreit 1986) CIG is based on the idea that constructions are represented as *partial information structures* which can be combined to build up larger structures. CIG differs from most recent grammatical theories in a number of ways.

The first major distinction of CIG is the ability to define constituents of constructions *semantically* as well as syntactically. CIG allows a constituent of a construction to be defined by any set of informational assertions, phonological, syntactic, semantic, or pragmatic. Thus semantic constraints on a constituent are part of the *definition* of a construction. If an instance of a construction violates semantic constraints on its constituents it is uninterpretable. §3.4 will describe constructions like the HOW-SCALE construction which require semantic information to correctly specify their constituents.

A second novel feature of CIG is the use of *weak constructions*. Weak constructions are abstract constructions, like the lexical weak construction *Verb*, which augment the representation of standard constructions by abstracting over them in an *abstraction hierarchy*. Weak constructions are used in the grammar for two purposes. First, they serve to structure the grammar by linking together strong constructions in a way that is useful for access, for creating new constructions, and for learning. Second, having weak constructions allows the grammar to specify an equivalence-class of constructions which can be used by a particular construction to constrain its constituents. In general CIG emphasizes the use of mechanisms which *structure* the grammar and disallows mechanisms such as lexical rules, metarules, or derivational rules which *derive* rules or structure within the grammar. Weak and strong constructions are discussed in detail in §3.7.

1.3.2 The Interpreter

The interpreter Sal builds an interpretation for a sentence by accessing grammatical constructions, integrating them together to produce multiple candidate interpretations, and then selecting a most-favored interpretation from among these candidates.

Sal's architecture consists of three components: the *working store*, the *long-term store*, and the *interpretation function*. The *working store* holds constructions as they are accessed, and partial interpretations as they are being built up. The *long-term store* holds the linguistic knowledge of the interpreter (i.e., the grammar). The *interpretation function* includes the access, interpretation, and selection functions. Figure 1.5 shows an outline of the architecture.

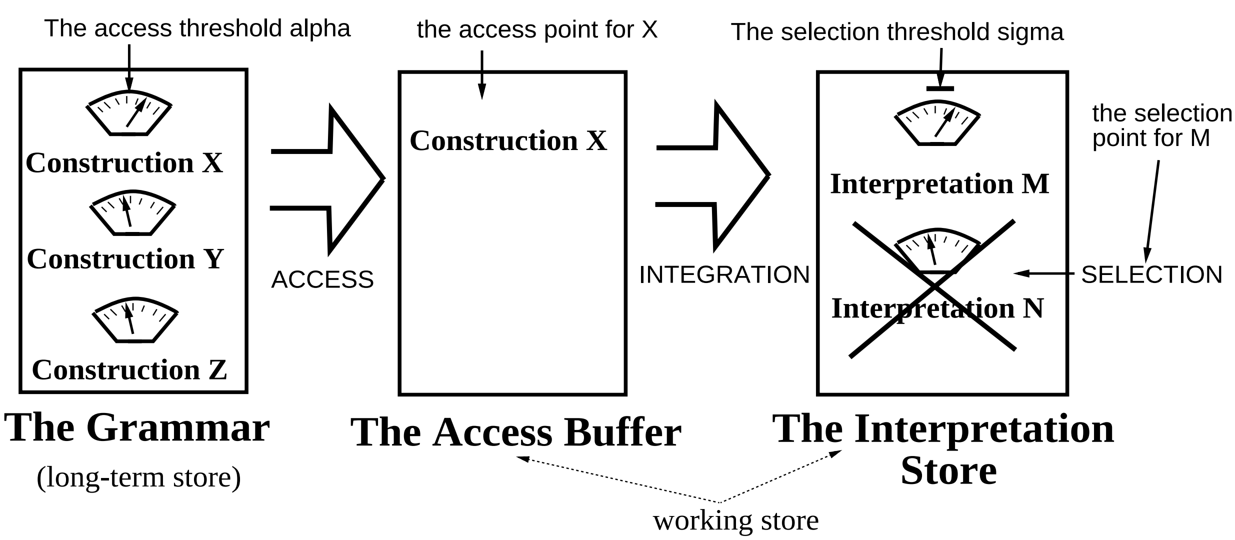


Figure 1.5: The Architecture of Sal

The first of the three sub-functions of the interpretive mechanisms is the *access function*.

Access Function: Access a construction whenever the evidence for it passes the access threshold α.

The *access function* amasses evidence for constructions that might be used in an interpretation. When the evidence for a construction has passed the *access threshold* α, the interpreter copies the construction into the **access buffer**, which is part of the *working store*. The ability to use different kinds of evidence, including syntactic, semantic, and frequency evidence, amassed from both bottom-up and top-down sources, makes this access function a much more general one than have been used in previous parsers or interpreters. Previous models have generally relied on a single kind of information to access rules. This might be *bottom-up* information, as in the shift-reduce parsers of Aho & Ullman (1972), or *top-down* information, as in many Prolog parsers, solely *syntactic* information, as in the left-corner parsers of Pereira & Shieber (1987), Thompson *et al.* (1991), and Gibson (1991), or solely *semantic* or *lexical* information, as in conceptual analyzers like Riesbeck & Schank (1978) or in Cardie & Lehnert (1991) or Lytinen (1991). The access algorithm presented here can use any of these kinds of information, as well as *frequency* information, to suggest grammatical constructions, and thus suggests a more general and knowledge-based approach to the access of linguistic knowledge.

> **Integration Function:** An interpretation is built up for each construction as each of its constituents is processed, by integrating the partial information provided by each constituent.

The *integration function* incrementally combines the meaning of a construction and its various constituents into an interpretation for the construction. The operation used to combine structures is also called *integration*, designed as an extension of the *unification* operation. While unification has been used very successfully in building *syntactic* structure, extending the operation to building more complex *semantic* structures requires three major augmentations:

- The integration operation includes knowledge about the representation language which is used to describe constructions (see §3.8).
- The integration operation distinguishes *constraints* on constituents or on valence arguments from *fillers* of constituents or valence arguments.
- The integration operation is augmented by a *slash* operator, which allows it to join semantic structures by embedding one inside another, in a similar way to the *functional-application* operation used by other models of semantic interpretation.

The *selection function* chooses an interpretation from the set of candidate interpretations in the interpretation store. The function chooses the interpretation which is the most *coherent* with grammatical expectations, according to the Selection Choice Principle:

> **Selection Choice Principle:** Prefer the interpretation whose *most recently integrated element* was the most *coherent* with the interpretation and its lexical, syntactic, semantic, and probabilistic expectations.

Selection is timed in an on-line fashion — the selection function *prunes* an interpretation whenever it becomes much worse than the most-favored interpretation in the interpretation store, according to the Selection Timing Principle:

> **Selection Timing Principle:** Prune interpretations whenever the difference between their ranking and the ranking of the most-favored interpretation is greater than the selection threshold σ.

1.4 A Sample Trace

In order to develop and test the model of interpretation, I have built a Common Lisp implementation of Sal, as well as a small CIG test grammar of about 50 constructions. This section presents a trace of the interpretation of the sentence “How can I create disk space?”. Further details of the processing of this sentence are presented in §4.9.

```
<cl> (parse '(how can i create disk space))

*** ACCESS ***
Input word: how
Bottom-up Evidence for constructions (means-how howscale)
Constructions (means-how howscale) are accessed
Bottom-up Access of (whnonsubjectquestion), integrated directly into Access Buffer
Top-down Evidence for constructions (aux)
Access of constructions nil

*** INTEGRATION ***
After integration, Store contains: ((whnonsubjectquestion whnonsubjectquestion
whnonsubjectquestion))

*** SELECTION ***
After removing failed integrations, Store contains: (whnonsubjectquestion
whnonsubjectquestion)
```

Figure 1.6:

In the first part of the trace in Figure 1.6, the input word “how” supplies evidence for two constructions, MEANS-HOW and HOW-SCALE, which are then accessed. These constructions then supply evidence for the WH-NON-SUBJECT-QUESTION construction, and these are integrated together. At the end of this stage, the interpretation store contains two WH-NON-SUBJECT-QUESTION interpretations, one with the MEANS-HOW construction and one with the HOW-SCALE construction. Note that there was some top-down evidence for the AUX construction, because the second constituent of the WH-NON-SUBJECT-QUESTION construction is constrained to be an AUX.

Figure 1.7 shows the second part of the trace, in which the input “can” provides evidence for the three lexical constructions CAN-1, CAN-2, and CAN-3, as well as some larger constructions, the DOUBLE-NOUN construction, and the BARE-MONO-TRANS-VP construction. Sal then attempts to integrate each of the two previous interpretations with these 5 constructions, as well as with the actual input word “can”, producing 12 possible interpretations. Most of these 12 interpretations are ruled out because they failed to integrate successfully, leaving only one. This successful interpretation includes the MEANS-HOW construction and the auxiliary sense of “can”.

In Figure 1.8 the word “i” is input and integrated into the interpretation. Note that although there is some top-down evidence for the verb-phrase construction, it is not accessed, because it is a *weak construction*, and there is insufficient evidence for any of the related *strong constructions*. Weak and strong constructions will be discussed in §3.7.

```
*** ACCESS ***
Input word: can
Bottom-up Evidence for constructions (can-1 can-2 can-3 doublenoun bare-mono-trans-vp)
Top-down Evidence for constructions nil
Access of constructions nil

*** INTEGRATION ***
After integration, Store contains: ((whnonsubjectquestion whnonsubjectquestion
whnonsubjectquestion whnonsubjectquestion whnonsubjectquestion whnonsubjectquestion
whnonsubjectquestion whnonsubjectquestion whnonsubjectquestion whnonsubjectquestion
whnonsubjectquestion whnonsubjectquestion))

*** SELECTION ***
After removing failed integrations, Store contains: (whnonsubjectquestion)
```

Figure 1.7:

```
*** ACCESS ***
Input word: i
Bottom-up Evidence for constructions (i)
Top-down Evidence for constructions (verb-phrase)
Access of constructions nil

*** INTEGRATION ***
After integration, Store contains: ((whnonsubjectquestion whnonsubjectquestion))

*** SELECTION ***
After removing failed integrations, Store contains: (whnonsubjectquestion)
```

Figure 1.8:

Next, in Figure 1.9, the word "create" is input and integrated into the interpretation, along with an appropriate type of verb-phrase.

In Figure 1.10 the word "disk" is input, which provides evidence for the lexical construction DISK, as well as the noun-compound DISK-SPACE, and the DOUBLENOUN construction.

Finally, in Figure 1.11 the word "space" accesses the lexical construction SPACE. The selection algorithm must now choose between two interpretations, one with the DISK-SPACE construction, and one with the DOUBLENOUN construction in which the nouns are respectively "disk" and "space". Because the DISK-SPACE construction has a strong expectation for the word "space", this first interpretation is selected. See §4.9 for further details.

```
*** ACCESS ***
Input word: create
Bottom-up Evidence for constructions (create bare-mono-trans-vp)
Top-down Evidence for constructions (noun-phrase)
Access of constructions nil

*** INTEGRATION ***
After integration, Store contains: ((whnonsubjectquestion whnonsubjectquestion
whnonsubjectquestion))

*** SELECTION ***
After removing failed integrations, Store contains: (whnonsubjectquestion)
```

Figure 1.9:

```
*** ACCESS ***
Input word: disk
Bottom-up Evidence for constructions (disk diskspace doublenoun)
Top-down Evidence for constructions (noun)
Access of constructions nil

*** INTEGRATION ***
After integration, Store contains: ((whnonsubjectquestion whnonsubjectquestion
whnonsubjectquestion whnonsubjectquestion))

*** SELECTION ***
After removing failed integrations, Store contains: (whnonsubjectquestion
whnonsubjectquestion)
```

Figure 1.10:

```
*** ACCESS ***
Input word: space
Bottom-up Evidence for constructions (space doublenoun)
Top-down Evidence for constructions nil
Access of constructions nil

*** INTEGRATION ***
After integration, Store contains: ((whnonsubjectquestion whnonsubjectquestion
whnonsubjectquestion whnonsubjectquestion whnonsubjectquestion whnonsubjectquestion))

*** SELECTION ***
After removing failed integrations, Store contains: (whnonsubjectquestion
whnonsubjectquestion)
Pruning construction 'whnonsubjectquestion' (1 points), because difference from
 construction 'whnonsubjectquestion' (3 points) exceeds selection threshold

Input Exhausted. Result is:
((a  question  $q
     (queried $p*)
     (background
      (a  means-for  $newvar285
          (means $p*)
          (goal (a  ability-state  $as
                    (actor (a  speechsituation-speaker  ))
                    (action
                     (a  forcedynamicaction  $newvar291
                         (action
                          (a  creation-action
                              (created (a  disk-freespace  ))
                              (creator (a  speechsituation-speaker  )))))))))))
```

Figure 1.11:

1.5 Overview of the Thesis

Chapter 2 discusses the relationship between the *grammar* and the *interpreter*, defining the role that CIG plays in Sal. It touches on issues of grammaticality and interpretability, as well as proposing the *Interpretive Hypothesis* as an alternative to the competence-performance distinction. Chapter 3 introduces CIG, giving a definition for the grammatical construction and a summary of the notations and mechanisms that are used to define it, including the concept of *valence*. Chapter 4 gives an overview of Sal, and defines the *access*, *integration*, and *selection* theories. Chapter 4 also summarizes the psycholinguistic evidence which bears on the interpreter, and includes a trace of the interpretation of a complex sentence, showing how the interpreter handles sentences with *wh*-elements. Chapters 5–7 present the details of the interpretation mechanism. Chapter 5 discusses the access theory, and shows how the access of a construction can be informed by many different kinds of linguistic knowledge. Chapter 6 gives the details of the integration theory, showing how the integration operation handles complex combinations like those caused by long-distance dependencies. Chapter 7 describes the selection theory, including the *means* by which interpretations are chosen as well as the *timing* of the choice.

Related models of interpretation are discussed in the previous-research sections of each chapter. Thus Chapter 4 discusses related interpreter architectures, and Chapter 5 discusses the access theories of a number of parsers and interpreters. Chapter 6 summarizes previous models of integration, including a discussion of information-combination operators as well as the granularity of integration, while Chapter 7 discusses previous models of selection *choice* and *timing*.

Finally, Chapter 8 summarizes problems with this work, and gives directions for future research.

Chapter 2

The Role of Grammar in Interpretation

Proposing a model of sentence interpretation consistent with both psycholinguistic data and linguistic criteria requires a reanalysis of the relation between *grammar* and *interpretation*. This chapter describes how this relation differs in the model proposed in this dissertation from the traditional generative-derivational model.

The familiar generative model of language distinguishes sharply between *competence* and *performance*. Chomsky (1965) defines linguistic *competence* as being concerned with

> an ideal speaker-listener, in a completely homogeneous speech-community, who knows its language perfectly and is unaffected by such grammatically irrelevant conditions as memory limitations, distractions, shifts of attention and interest, and errors (random or characteristic) in applying his knowledge of the language in actual performance.

In this sense, competence acts as what Derwing (1973) has called ‘an idealized model of linguistic performance’. Such idealizations are essential for a model of interpretation like Sal. As Gibson (1991) notes, such factors as shifts of attention and interest, for example, are independent of linguistic processing, and can be better explained by non-linguistic psychological models.

While the importance of this aspect of competence as an *idealized* model of language use is undeniable, in practice linguistic competence has also acquired a status as an *autonomous subsystem* of a model of language. Thus the derivational model of language consists of four components; two related to *competence* and two to *performance*:

[1a] A list of rules comprising a *competence grammar* of the language.

[1b] A *competence derivational algorithm*, which follows these rules to create parse trees for sentences.

[2a] A *performance grammar*, which corresponds in some unspecified fashion to the competence grammar (see §2.1 on the nature of this correspondence).

[2b] A set of *performance algorithms* which accounts for interpretation, production, and learning by applying this performance grammar.

In other words, the traditional model includes a distinct and autonomous *competence* component, which includes both a list of grammatical rules and a derivational algorithm which follows these rules and builds phrase-structures. Chomsky (1972) stated quite clearly that although he characterized linguistic competence as a "system of processes and rules", this "system" need not be related to the processes and rules which speakers of a language use to build interpretations:

> The [perceptual model] PM incorporates the grammar G of a language. . . But it is important to distinguish clearly between the function and properties of the perceptual model PM and the competence model [grammar] G that it incorporates. . . Although we may describe the grammar G as a system of processes and rules that apply in a certain order to relate sound and meaning, we are not entitled to take this as a description of the successive acts of a performance model such as PM — in fact, it would be quite absurd to do so. (p. 117)

The model described in this dissertation proposes the **Interpretive Hypothesis**, which redraws this competence-performance distinction, resulting in a system with only two components instead of four. The interpretive hypothesis proposes that the two components of a theory of language are

1. A Construction-Based Interpretive Grammar (CIG), consisting of a collection of declarative grammatical constructions.

2. A set of procedures which model interpretation, production, and learning. This dissertation only discusses the first of these, the interpretation procedure Sal.

The traditional four components are reduced to two by eliminating two parts of the derivational model. First, the interpretive hypothesis removes the distinction between the *competence rule-base* and the *performance rule-base*, resulting in only a single collection of grammatical rules, with a single functional role as a structural ingredient in processing.

Second, the model does not include any *competence derivational algorithm*. Thus CIG is *non-constructive* in the sense of Langendoen & Postal (1984), in that the grammar does not model language by *constructing* structures for sentences. Rather, structure is built by the processing component of the model, the interpreter Sal. Note that the distinction between a *rule*, in the derivational or constructive sense, and a *construction*, in our sense, is that a rule implies the existence of a derivational mechanism which *follows* the rule.

Recasting the model of language in this way augments linguistic competence to include *language in use* or *language processing* as part of linguistic theory, resulting in a much tighter relation between the grammar and the interpreter than in the derivational model. The interpretive hypothesis still assumes that the model is an idealization, ignoring such factors as attention shifts. And the non-derivational model will still address many of the same problems as a derivational model. Thus in building a correct interpretation for a sentence, the interpreter will be demonstrating that there is such an interpretation for any sentence in a language. But significantly, the goal of this theory, modeling the interpretation of sentences, is very distinct from this traditional generative one of "generating all and only the sentences of a language"

The rest of this chapter will discuss the implications of removing two components from the model:

- §2.1 summarizes the implications of collapsing the *performance* rule-base and the *competence* rule-base. Obviously this means that the two rule-bases cannot be distinct; a corollary of this is that the theory does not allow a class of sentences which are *grammatical but not acceptable*.

- §2.2 discusses the results of removing the *competence derivational algorithm* from the model. This means that grammatical knowledge is solely *declarative* — any structure-building is performed by the interpretive, production, or learning mechanisms. This rules out derivational algorithms, grammatical transformations, as well as lexical rules. Thus CIG is *non-constructive*.

2.1 No Distinct Competence Grammar

The first distinction between the derivational model and the interpretive model proposed here is that the interpretive model does not allow a distinction between the *competence grammar* and the *performance grammar* of a language. The model only has a single linguistic knowledge component, represented as a collection of grammatical constructions. This model of grammar shares some features with the traditional derivational view. Like Chomsky (1965), this model assumes that "a generative grammar . . . attempts to characterize in the most neutral possible terms the knowledge of language by a speaker-hearer (p. 9)." That is, it assumes that these declarative knowledge structures abstract away from the details of interpretation or production. Furthermore, conflating the competence and performance grammars does not rule out such generative notions as developing methodological tools for investigating the structural portion of the model (such as grammatical intuitions). But significantly, there is only one such structural portion, with a single functional role as a structural ingredient in processing.

The interpretive hypothesis rules out any separate functional role for the grammar. That is, any mechanisms which are included in the grammar to account for particular linguistic phenomena must be visible to the processing mechanism. Thus mechanisms which are proposed to capture linguistic generalizations, for example, such as redundancy rules or transformations, are either present in the processing mechanism or are merely convenient metatheoretical or historical abstractions, and can thus have no causal role in a theory of human language processing. This makes the relationship between the grammar and the interpreter much tighter than in previous models.

Many derivational models of grammar have assumed some close relation between the competence grammar of a language and the rule base (or "representational basis") of a processing model. One of the earliest such formulations is by Miller & Chomsky (1963), who proposed that the language processing mechanism is "a finite device M in which are stored the rules of a generative grammar G" (p 466). Miller and Chomsky thus propose that the processing mechanism M contain precisely the same rules as the grammar G.

Miller and Chomsky's position was restated in a form that Bresnan & Kaplan (1982) have called the *competence hypothesis*, by Chomsky (1965):

> . . . a reasonable model of language use will incorporate, as a basic component, the generative grammar that expresses the speaker-hearer's knowledge of the language. . . (p. 9)

The competence hypothesis proposes that the competence and performance grammars be closely related by "incorporation". This is similar to our requirement that the competence and performance grammars be identical, but differs in two ways from the interpretive hypothesis First, the competence hypothesis still maintained a distinct functional role for the competence grammar; it was a separate entity, used as part of a device to enumerate grammatical sentences, and as such has its own role in the language faculty. Second, the notion of *incorporate* was left quite vague. At some times in his formulation of the idea, Chomsky seemed to mean by *incorporate* that only the *representational* aspect of the generative grammar (i.e., the rule-list) would be incorporated, without the derivational or transformation algorithms (as in the quotation from Chomsky (1972) on page 18). At other times, he seemed to include as well the transformational algorithms and processes, as in the following passage from Miller & Chomsky (1963), which seems to argue for the Derivational Theory of Complexity:

> The psychological plausibility of a transformational model of the language user would be strengthened, of course, if it could be shown that our performance on tasks requiring an appreciation of the structure of transformed sentences is some function of the nature, number, and complexity of the grammatical transformations involved. (p. 481)

Neither of these positions is possible for us, since our theory not only disallows a difference between the competence and performance grammars, but allows only a single functional role for grammatical knowledge.

Our position closely resembles the *Strong Competence Hypothesis* of Bresnan & Kaplan (1982:xxxi), which requires that the representational basis of the process model be "isomorphic to the competence grammar." However, although the requirement of isomorphism disallows any grammatical rules which are in the competence grammar but not the performance grammar, Bresnan and Kaplan still allow a class of *lexical rules* which appear in the competence model but not in the performance model. As Bresnan & Kaplan (1982:xxxiii) note:

> such lexical rules, as long as they have a finite output, can always be interpreted as redundancy rules . . . As such, the rules could be applied to enter new lexical forms into the mental lexicon, and the derived lexical forms would subsequently simply be retrieved for lexical insertion rather than being rederived.

Thus in order to capture useful generalizations, the performance grammar of LFG is augmented by these "competence lexical rules".

CIG does not allow such lexical rules, since they would not be included in the performance grammar, and hence violate the interpretive hypothesis.[1] CIG replaces these rules with specific

[1] Although if these lexical rules were an active part of the grammar-learning mechanism, they would not violated the interpretive hypothesis

constructions; Lakoff (1977), Jurafsky (1988), Goldberg (1989), and Goldberg (1991) discuss how Construction Grammar uses constructions to represent phenomena traditionally handled by such redundancy rules (see also §3.7).

As a number of researchers in the Government and Binding paradigm have pointed out, there is no logical necessity that the competence and performance grammars be identified. Berwick and Weinberg (1983 and 1984), for example, argue that the relation between the two may be weaker than isomorphism; that the mapping may be non-surjective or non-injective, or both. Indeed, they claim that the grammars may be related by the even weaker notion of *covering*. Berwick & Weinberg (1984) informally characterize covering as follows:

> Informally, one grammar G_1 covers another grammar G_2 if (1) both generate the same language $L(G_1) = L(G_2)$, that is, the grammars are weakly equivalent; and (2) we can find the parses or structural descriptions that G_2 assigns to sentences by parsing the sentences using G_1 and then applying a "simple" or easily-computed mapping to the resulting output. (p. 79)

This model of covering grammars is used by many of the *Principle-Based Parsers*, such as Abney (1991), Johnson (1991), Fong (1991), and Correa (1991), which compile a number of the principles of GB to produce a covering grammar which is then used by the parser.

The interpretive hypothesis of course rules out such models, but it is interesting to note that these covering grammars resemble construction grammars much more than they do GB grammar: As Berwick (1991) remarks about a covering grammar, it

> is *not* pure X-bar theory — it actually looks more like a conventional context-free rule-based system. . .

Indeed, the *chunk-parser* of Abney (1991) uses a grammar which bears little if any relation to GB principles at all — his *chunks* are specifically defined as re-write rules, and when viewed declaratively bear a close resemblance to grammatical constructions.

The interpretive hypothesis is preferable to these models on the grounds of Occam's razor; the CIG model includes only a single grammar, where the GB model must include two. The fact that the performance grammars used by these parsers resemble construction grammars is additional evidence for the kind of grammar described by CIG.

2.2 No Derivational Algorithms

As the introduction to this chapter discussed, the *competence grammar* that was part of the generative-derivational theory of language included two components: a collection of rules, and an algorithmic procedure for following these rules and assigning structural descriptions to sentences. The two parts combined to make a competence grammar "a system of rules that in some explicit and well-defined way assigns structural descriptions to sentences" (Chomsky (1965:8)). In an earlier citation, Chomsky (1962) makes the algorithmic nature of this model even more explicit, defining the grammar for a language L as

> a device which enumerates the sentences of L in such a way that a structural description can be mechanically derived for each enumerated sentence. [Reprinted in Fodor & Katz (1964:240-1)]

The interpretive hypothesis rules out this second, algorithmic, part of the model. Knowledge of language is thus defined as knowledge of a declarative set of structures, and any principles which specify how these structures are *combined* must be part of the interpretive or productive mechanisms of the language faculty.

Clearly this view that a theory of language can be expressed solely by a collection of representations rather than by a set of rules is incompatible with many theories of grammar. The interpretive hypothesis disallows, for example, the transformations of *Aspects*-era generative grammar, the move-α of Government and Binding theory (Chomsky 1981), or the redundancy rules of Jackendoff (1975) and Bresnan (1982a). But more significantly, it excludes the most fundamental and pervasive non-declarative mechanism, one which is present in some form in most if not all modern theories of grammar, the *derivational rule system* of phrase-structure grammar.

This section will discuss two aspects of the lack of this derivational algorithm. First, §2.2.1 discusses how a non-derivational system can still capture the Humboldtian "creativity of language" which was an early inspiration of the generative model. §2.2.2 then discusses how a non-derivational theory models the derivational distinction between *grammatical* and *acceptable* sentences, proposing a new criterion: *interpretability*. Finally, §2.2.3 summarizes arguments by a number of scholars against derivational theories.

2.2.1 Derivation and Creativity

The derivational rule system was proposed by Chomsky (1956/1975 and 1957) to capture a notion of *process* that Chomsky claimed was missing from earlier American Structuralist models. Chomsky relied on two familiar ideas that he credited to Humboldt (discussed in Chomsky 1964 and 1966). The first is Humboldt's (1836/1988) comment that a theory of human language processing must account for the infinite creativity of language processing with finite means.

> . . . the procedure of language is not simply one whereby a single phenomenon comes about; it must simultaneously open up the possibility of producing an indefinable host of such phenomena, and under all the conditions that thought prescribes. For language is quite peculiarly confronted by an unending and truly boundless domain, the essence of all that can be thought. It must therefore make infinite employment of finite means, and is able to do so through the power which produces identity of language and thought. (p. 91)

This problem might be expressed as the constraint that a theory of human language processing must model the human ability to recognize and produce novel utterances. The second justification for the rule-system as a characterization of the human language faculty appealed to Humboldt's dictum that "[Language] in itself. . . is no product (Ergon) but an activity (Energeia)." (p. 49)

Chomsky's solution was to characterize knowledge of language by intension rather than by extension — that is, by using recursive function theory to define the subset of the set of strings of words which are members of the language without resorting to enumeration. The theory consisted of two components: a list of structures (i.e., phrase-structure rules) and, more importantly, a derivational algorithm which combined these structures (i.e., followed the rules). The algorithm, which rewrites structure from S to terminal symbols, was supposed to act as the *creative force* in a language model. Chomsky thus required that the *grammar* itself, the *representational* component of his theory of language, be "creative". Chomsky seems to have isolated creativity in the representational component because of his attempt to abstract away from processing details in making the competence-performance distinction (see Boas (1975) for a further discussion of this unusual definition of "creativity").

In the model of interpretation presented in this dissertation, the creative spirit of human language as a property of the language faculty as a whole, and not merely of the representational component. Thus the locus of "creativity" is in the cognitive processes which constitute the "processing ingredients" of the human language processing mechanism. That is, we constrain the processes which model the interpretation and production aspects of the human language mechanism to use linguistic structural knowledge in a creative way. Thus instead of saying that the grammar alone must generate 'all and only the sentences of English', we say that our model of language understanding as a whole must be able to interpret and produce the sentences of English, allowing us to dispose of the part of the generative mechanism that attempted to account for creativity solely in the grammar, the derivational algorithm. In removing this mechanism from the grammar we focus on the task of producing a cognitive model of human language interpretation or production instead of the task of listing all and only the grammatical sentences of English.

Indeed, this appeal to creativity as a function of language *use* can be seen in Humboldt as well:

> . . . language resides in every man in its whole range, which means, however, nothing else but that everyone possesses an urge governed by a specifically modified, limiting and confining power, to bring forth gradually the whole of language from within himself, or when brought forth to understand it, as outer or inner occasion may determine.

2.2.2 No Grammaticality

Because the generative-derivational model of grammatical competence includes a derivational algorithmic component which is distinct from the performance interpretation mechanism, it may build up structure in a different way than the performance model. This difference may allow the performance model to accept a different set of sentences than the performance model.

In fact Miller & Chomsky (1963) argue that the processing mechanism may accept a *subset* of the sentences which are accepted by the competence grammar:

> We say that the device *M (partially) understands the sentence x in the manner of G* if the set $\{F_1(x), \ldots F_m(x)\}$ of structural descriptions provided by M with input x is (included in) the set assigned to x by the generative grammar G. (p. 466)

That is, the device M accepts some subset of the grammatical sentences accepted by G.

Miller & Chomsky (1963) claim that the reason that the grammar G will accept some sentences which the processing model M will reject is that it is possible that "M will not contain enough computing space to allow it to understand all sentences in the manner of the device G" (p 467). In other words, the processing model is bounded by memory limitations which do not apply to the competence grammar G. Miller and Chomsky proposed that it is these memory limitations which make nested dependencies or self-embedding structures in natural language difficult to interpret. In other words, Miller & Chomsky (1963) proposed that deeply center-embedded constructions are *grammatical*, they simply are not able to be assigned a structure by the human language interpretation mechanism.

Chomsky (1965) extends this discussion by giving a name to this class of sentences, those which may be grammatical but to which it is difficult to assign a structure. Chomsky calls these sentences “unacceptable”, and gives as an example (2.1), which he considers grammatical but unacceptable:

(2.1) The man who the boy who the students recognized pointed out is a friend of mine.

The tradition in generative grammar, then, has been to label sentences which cause processing overloads such as the ‘center-embedded’ structures mentioned above, or the garden-path sentences first noted by Bever (1970) as *grammatical* but *unacceptable*.

But how is the theorist to decide that these sentences are grammatical? The only source for grammaticality judgments that the theory allows, native speaker grammaticality intuitions, certainly do not accept sentences like (2.1) above. Rather, the claim that sentences which are unacceptable due to processing limitations are nonetheless grammatical is a claim that Miller & Chomsky (1963) must make by fiat (this point is discussed by Reich (1969)).

Figure 2.1 uses Venn diagrams to show the generative-derivational model of grammaticality. The outer circle contains the set of grammatical sentences, while the inner one contains the set of acceptable sentences. Thus the disjunction, those sentences in the outer set which are not in the inner set, are the sentences like (2.1) which are grammatical but not acceptable. Besides the center-embedded sentences described by Chomsky, most recent generative computational linguistics assume that garden-path sentences such as *“The horse raced past the barn fell”* are also included in the category.

The interpretive model does not assume this set of *grammatical-but-unacceptable* sentences, because it does not model a language by generating sentences like (2.1), and then ruling them out by an acceptability filter. Rather, the interpretive model describes a language by describing which sentences are *interpretable*. A sentence is *interpretable* if the interpreter is able to successfully assign it an interpretation. Under this definition, the garden path sentences such as (2.1) are *uninterpretable*, meaning that the interpreter is unable to process them without appealing to some higher-level reasoning capacity.

Although the model does not build *sentences* which are grammatical but unacceptable, it can distinguish sentences which are *not* interpretable for grammatical reasons from those which are not interpretable for processing reasons. Thus a *grammatically uninterpretable* sentence is one which is uninterpretable because of syntactic or other grammatical reasons, while a *process uninterpretable* sentence is one which is uninterpretable for processing reasons. The difference is that the interpretive model does not build and then filter out structures for these sentences as the derivational model does.

In conclusion, the interpretive hypothesis leads to a notion of a linguistic theory as an idealization of *language processing*, which includes both a collection of declarative structures constituting *knowledge of language* and a set of processing functions constituting the interpretive, productive, and learning aspects of human language use.

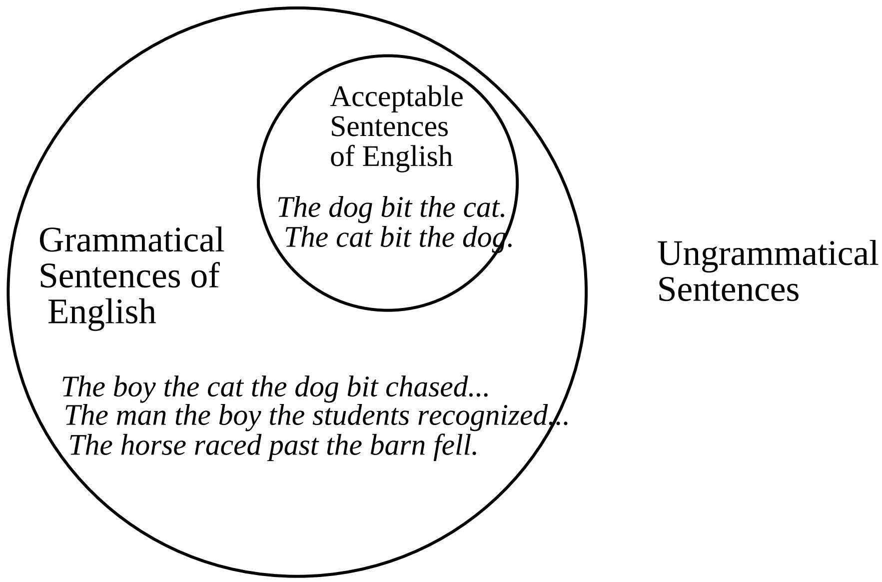


Figure 2.1: The Generative Grammar Definition of Grammaticality and Acceptability

2.2.3 Other Arguments Against Derivation

A number of researchers have advanced arguments against a derivational or 'constructivist' theory of grammar. Langendoen & Postal (1984), for example, have shown that the set of sentences of a natural languages is too large to be recursively enumerable. Because of this, grammars which model language by *constructing* the set of all possible sentences are insufficient to characterize the complete set. However Langendoen & Postal's (1984) argument is based on a consideration of sentences of infinite length, which is only possible because they do not consider the problem of *interpretation*.

A number of scholars have also argued that a derivational model of grammar is insufficient because the *meaning* of particular sentences is dependent on their being constructed by an *interpretive model*, and not by a *derivational algorithm*. For example, Fish (1970) has studied a number of examples where the *meaning* of an utterance can only be expressed in the context of a *temporally-embedded* model; i.e., examples where the meaning of a sentence is dependent on the fact that sentences are interpreted on-line. Fish considers examples where factive clauses are interpreted differently depending on the timing of their occurrence in the sentence.

Wilensky (personal communication) and Norvig (1988) have advanced similar arguments against constructive theories based on what Wilensky has called "semi-double entendres". In these cases, sentences such as the newspaper headlines *"Bears maul Tigers"* or *"Dairy industry sours"* have a coherent interpretation based on the metaphorically extended senses of the words, but are designed so as to encourage a second reading based on the central sense of the words, which has a humorous effect. This second interpretation (based on, e.g., the link between *dairy* and the non-metaphorical sense of *sours*) is necessarily part of the meaning of the utterance, but it is difficult to imagine how a derivational grammar would build this second interpretation. An interpretive model, on the other hand, could certainly build both of these interpretations; theories of interpretation such as Hirst (1986) and Waltz & Pollack (1985) have modeled similar effects by the use of *spreading activation* techniques.

Chapter 3

Representation of Linguistic Knowledge

3.1 Introduction

This chapter introduces and defines Construction-Based Interpretive Grammar (CIG), the theory of grammar that is used by the interpretation model Sal. The chapter gives a detailed exposition of the grammar and its principles, including the definition of the representation language which will be used for grammatical examples throughout the dissertation.

CIG derives from Construction Grammar (Fillmore *et al.* 1988; Kay 1990; Lakoff 1987) which proposes a return to the traditional notion of the grammatical construction by:

> describing the grammar of a language directly in terms of a collection of *grammatical constructions* each of which represents a pairing of a syntactic pattern with a meaning structure. . . (Fillmore (1989b:4))

CIG is based on three fundamental principles which describe the form of the grammar and its role in the model of sentence interpretation.

> **Linguistic Knowledge Principle:** A single uniform collection of grammatical constructions is the sole representation of linguistic knowledge, and is the structural ingredient of all aspects of human language processing, interpretation, production, and learning.

The **Linguistic Knowledge Principle** outlines the nature of the grammar. A CIG grammar consists of a declarative collection of structures called *grammatical constructions* which resemble the constructions of traditional pre-generative grammar.[1] Each of these constructions represents information from various domains of linguistic knowledge; phonological, syntactic, semantic, and pragmatic knowledge. Thus the grammar constitutes a database of these constructions, which might be called a *constructicon* (on the model of the word *lexicon*).

The principle makes two further claims. First, the grammar consists *solely* of a collection of grammatical constructions. That is, the grammatical construction is a sufficient representational primitive to completely characterize the grammar of a language, without additional rule types, such as derivational or redundancy rules. Second, this grammar is the structural ingredient in each aspect of human language processing, including interpretation, production, and learning.

> **Uniformity Principle:** The constructions of the grammar uniformly represent linguistic structures from the simplest to the most complex, whether lexical, idiomatic, or syntactic.

The Uniformity Principle claims that lexical entries, idioms, and syntactic structures are all represented uniformly as grammatical constructions. Thus the "constructicon" subsumes the lexicon, the syntactic rule base, and the idiom dictionary assumed by other theories. Constructions vary in size from lexical constructions, which represent knowledge of the phonology, syntax and semantics of lexical items, to large clause- or sentence-level constructions like the YES-NO-QUESTION or WH-SUBJECT-QUESTION constructions. Using a single representation for linguistic knowledge allows a very general mechanism for language understanding — lexical access, idiom processing, syntactic parsing, and semantic interpretation are all done by the same mechanism using the same knowledge base.

Most models of language structure represent different types of linguistic knowledge in very disparate ways. Traditional models collect lexical entries into a lexicon, idioms into an idiom

[1] The idea of the grammatical construction as a representational unit in a grammar was assumed by the American Structuralists, and can even be found in Saussure. For example, Saussure (1915/1966:125) notes that "To language [langue] rather than to speaking [parole] belong the syntagmatic types that are built upon regular forms [i.e., constructions]. . . the same is true of sentences and groups of words built upon regular patterns. Combinations like *la terre tourne* 'the world turns,' *que vous dit-il?* 'what does he say to you?' etc. correspond to general types that are in turn supported in the language by concrete remembrances." Although a number of scholars maintain that on the contrary Saussure considered syntax to be part of *parole* (see Chomsky (1965), Pollard & Sag (1987), Sampson (1980)), it is quite possible this is based on a misreading of Saussure's reference to the 'sentence' as part of *parole*, ("The sentence is the ideal type of syntagm. But it belongs to speaking, not to language."(p 124)), but it seems clear that by 'sentence', Saussure means 'utterance', and not the more abstract epistemological notion of sentential construction.

dictionary, syntactic rules in a phrasal rule-base, and semantic rules in a semantic rule-base. Obviously unifying these has the advantage of simplicity. Appealing to Occam's Razor, a single linguistic knowledge base which can represent lexical, idiomatic, and syntactic knowledge is more economical and efficient than separate knowledge-bases with separate representational mechanisms. But in addition, a unified representational mechanism allows us to build a theory of interpretation that applies all kinds of evidence in the construction of an interpretation in an on-line, flexible, and integrated fashion. This means that knowledge can be *accessed* uniformly — Chapter 5 shows how lexical access, syntactic access, idiom access and semantic rule access can be modeled with a single mechanism. Second, knowledge can be *integrated* uniformly — Chapter 6 shows how information from these multiple knowledge sources can be easily integrated in an on-line and cognitively plausible manner in producing an interpretation of an utterance. Finally, interpretations can be *selected* on the basis of coherence with expectations derived from any of these types of linguistic knowledge, as described in Chapter 7.[2]

> **Information Principle:** Each construction of the grammar may represent phonological, syntactic, semantic, and constructional information.

The Informational Principle describes the kind of information that is represented in each construction. Each construction may include knowledge about any domain of linguistic knowledge, including syntactic, semantic, phonological, pragmatic, and frequency information. The set of predicates which define a construction and its constituents can be viewed as a set of constraints on possible instances of the construction.

The idea that a *lexical item* can be represented as a pairing of meaning and form is of course directly in the tradition of Saussure. But construction grammar extends this approach to larger, non-lexical constructions such as the PASSIVE construction or the SUBJECT-PREDICATE construction. In allowing us to speak of the meaning of a construction just as it is possible to speak of the meaning of a word, the grammatical construction resembles the representational primitives of a number of recent theories, beginning with Montague (1973) and including Becker (1975), Lakoff (1977), Bolinger (1979), Wilensky & Arens (1980), Zwicky (1987), and Zwicky (1989).

The grammatical construction is defined as a complex pairing of meaning and form, which means that a construction is a relation between two *information structures*, rather than a relation between form and meaning. These information structures can consist of syntactic knowledge, semantic knowledge, or both. Thus, whereas the *sign* expressed a relation between a set of ordered phonemes and a meaning structure, the construction abstracts over this by replacing 'ordered sets

[2]Since construction grammars like CIG remove the distinction between the lexicon and the grammar, does any interesting distinction remain between lexical and higher-level structures? Is it still possible to talk about "lexical constructions"? One might imagine drawing a distinction between those structures that make recourse to phonological information and those that do not. This, however, would give us the wrong intuition for constructions like the HOW-SCALE construction of §3.4, which contains phonological specifications like a lexical entry, but also specifies two constituents and their ordering. Similarly, grammatical idioms like "kick the bucket" or "bury the hatchet" contain phonological information but also allow verbal inflection and other grammatical modifications. (See Fillmore (1978) on these issues). An alternative possibility is to draw the line between constructions with multiple constituents and those with single constituents. But many non-lexical constructions are likely to have a single constituent. It seems likely that there might be no completely satisfactory demarcation between lexical and non-lexical constructions. Of course the essential issue is that CIG requires no such demarcation. This issue is discussed further in §3.7.

of phonemes' with abstractions over them. These abstractions can be syntactic or semantic ways of expressing more abstract categories to which these phoneme sequences belong. The construction is thus a part-whole structuring relating these categories. From a learning perspective, these constructions might be viewed as progressively more complex abstractions over pairings of phoneme-strings and representations of real-world situations. As the learner's grammar grows, it grows in complexity from a collection of lexical-style members involving simple form-meaning correspondences, to a richer collection of more complex and structured relations between abstract linguistic categories and relevant semantic features of the representation of situations.

Further details of the form-meaning relationship in the grammatical construction are discussed in §3.4, where the notion of *semantically-constrained constituents* is defined.

We conclude this section with a statement of the formal definition of the grammatical construction. Following the intuitions of Fillmore *et al.* (1988), a grammatical construction is proposed by the grammar-writer whenever there is a need to express some non-compositional aspect of meaning; a construction's *constituents* are proposed whenever they are syntactically or semantically required to express the correct definition of the construction:

The Grammatical Construction: A construction c may be defined with a group of constituents g if and only if

- There is some linguistic information associated with the construction c which is not predictable from the information associated with each of the individual constituents g_i, *and*
- Each of the individual constituents g_i is required to be in the construction because:
  - Writing the semantic form of the *constitute*[3] of c requires including the constituent g_i in order to build the correct interpretation, *or*
  - The constituent g_i is obligatory, in the sense that the construction c may not appear without the constituent g_i

The rest of this chapter will define and motivate this concept of grammar, and the grammatical construction itself. First, however, §3.2 will attempt to summarize the differences and similarities between CIG and other recent computational grammatical theories. §3.3 will provide a description of the representational notation for a simple lexical construction. §3.4–§3.6 describe the representation of more complex constructions, especially those which make use of construction *constituency*. §3.7 introduces the idea of *strong constructions* and *weak constructions*, and shows how the relation between these constructions can be represented using an *abstraction hierarchy*.

3.2 Related Theories of Grammar

CIG is among the many modern linguistic theories which have abandoned the *derivational* metaphor which permeated early generative grammar. This use of *declarative structures* as a theoretical foundation, as opposed to the metaphors of *process* used by the generative approach to grammar, in many ways constitutes a return to pregenerative American Structuralist positions. Of course, the need for declarative rather than procedural models was discussed early in the

[3]The constitute is the semantic rule built into a construction. See page 35 for a more complete definition.

century by Saussure (1915/1966), who emphasized the importance of avoiding terminology which "suggests a false notion of movement where there is only a state (p. 160)". Saussure's disagreement with procedural models came from his attempt to build synchronic models which did not use diachronic process-oriented terminology. Similarly Hockett (1954:386) noted a shift in late American Structuralism toward more declarative models "at least in part because of a feeling of dissatisfaction with the 'moving-part' or 'historical' analogy" (but cf. Hymes & Fought (1981)).

This structuralist-inspired view of linguistic knowledge has made significant inroads in modern theories of grammar. Most of these theories fall into one of two classes, each of which derives from an important early model. The first class, the *unification-based* theories, is based on Kay's (1979) seminal work on functional grammar. These theories, such as LFG (Bresnan 1982a) and LFG-based theories like that of Fenstad *et al.* (1985), HPSG (Pollard & Sag 1987), and recent versions of categorial grammar such as Uszkoreit (1986) and Karttunen (1989) represent linguistic information by *partial information structures* which are integrated in language processing by the *unification* operation (see Shieber (1986) for an introduction and survey). The use of partial information and the monotonic unification operation commit these theories to declarative representations by placing no constraints on the processing method.

The second class of theories includes the Government and Binding theory of Chomsky (1981), as well as a number of its offshoots, including the computational implementations of the theory, known as *principle-based parsers* (Berwick 1991) because they build structures using a small set of universal *principles* derived from Government-Binding theory rather than a set of phrase-structure rules. The principle-based theories derive their declarative metaphor from McCawley's (1968) proposal that phrase-structure rules be interpreted as a set of declarative *admissibility constraints*.[4]

CIG and other versions of Construction Grammar are closely related to the first class of theories above, especially LFG and HPSG. The rest of this section sketches a number of broad similarities and differences between CIG and these theories, touching only sketchily on the larger differences between CIG and the principle-based theories. We begin by comparing the *information-theoretic* level of the theories, and then describe the nature of the grammars themselves.

CIG uses as its basic informational representation a semantic-network-like language (to be described in §3.8.1) which represents partial information structures. Thus the language is a notational variant of the feature structures used by feature unification models such as Pollard & Sag (1987), Bresnan (1982a), and Uszkoreit (1986). Unlike the *term structures* of definite-clause grammar models (Pereira & Shieber 1987), which are essentially a subset of feature-structures, the language is not position-dependent, but attribute-dependent. That is, predicates are represented by *concepts* which have labeled slots. Like both feature and term unification, information structures in CIG are represented by variables, and combining structures occurs by binding together variables in different structures.

Although in these respects the representation language resembles the feature structures used by feature-unification, the primitive information-combining operator is not unification but an extension of unification called *integration*. It is described in Chapter 6.

At the grammatical level, CIG and other construction grammars are distinctive in their em-

[4] Although McCawley (p. 247) credits Richard Stanley (pers. comm. 1965) with the idea — a similar idea is certainly present in Stanley (1967).

phasis on semantic and pragmatic knowledge. In some ways, this use of semantics resembles other modern theories, particularly the unification-based theories such as LFG (Bresnan 1982a) and HPSG (Pollard & Sag 1987). Like LFG, CIG expresses complement selection mainly at the level of predicate argument structure (Grimshaw 1979; Bresnan 1982b), but unlike LFG, it allows semantic and syntactic constraints to be placed on the same complement in a uniform fashion. Also like LFG and HPSG, CIG and other construction grammars allow no enrichment of surface form. This rules out traditional syntactic devices such as traces, syntactic gaps, or syntactic coindexing. CIG's assumption that the *constitute* of each construction specifies how the interpretation is constructed resembles the semantic interpretation rules which augment syntactic rules in Montague (1973) and related formalisms, and in GPSG (Gazdar 1981; Gazdar 1982; Gazdar *et al.* 1985).

In other ways, CIG's emphasis on semantics distinguishes it from LFG or HPSG. For example, each CIG construction may have semantic or pragmatic properties. Neither LFG nor HPSG allows constructions to specify particular pragmatic properties. They do allow lexical entries to have semantic properties, but do not extend this ability to larger constructions. These theories do not allow constructions to have semantic properties which are not predictable from the semantics of their constituents. A number of researchers outside of construction grammar have argued for the need for such complex partially non-compositional constructions, including Makkai (1972), Becker (1975), Zwicky (1978), Bolinger (1979), Wilensky & Arens (1980), and Gross (1984).

CIG's emphasis on semantics leads it to capture generalizations on the semantic level rather than the syntactic level whenever possible. This is especially noticeable in the treatment of long-distance dependencies. A distant element like a *wh*- construction is not related to an *empty category* or *trace*, and is not *syntactically coindexed*, as in GB. Nor is it related to grammatical relations like SUBJ and OBJ at the *functional* level, as in the *functional uncertainty* of LFG. Instead, CIG associates a distant element *directly with the semantics of the distant predicate*. This resembles the recent proposals in the *categorial grammar* framework of Pickering & Barry (1991). §3.6 discusses how some sentences with *wh*-elements are represented in the grammar, while §6.5 discusses how *wh*-elements are semantically integrated with their predicates by the *integration* operation of Chapter 6.

Because of the Interpretive Hypothesis discussed in Chapter 2, CIG does not allow derivational mechanisms proposed to capture generalizations which play no role in language processing. These include the c-structure rules of LFG, the redundancy rules of Jackendoff (1975) and Bresnan (1978), which are used in LFG as well as HPSG, or the metarules of GPSG. Lakoff (1977), Jurafsky (1988), Goldberg (1989), and Goldberg (1991) discuss how phenomena traditionally handled by redundancy rules and transformations can be represented as constructions. §3.7.4 relates redundancy rules to the inheritance hierarchy used in AI models.

The *weak constructions* of CIG resemble the inheritance hierarchies of HPSG — §3.7 discusses similarities and differences. CIG tends to use semantics to capture most of the kind of generalizations for which HPSG uses inheritance, reserving weak constructions for generalizing over cases where HPSG uses lexical rules.

Another major distinction between CIG and other theories concerns the role of *grammatical relations* or *functional roles*. Concepts like *subject* or *object*, which are primitives at the level of f-structure in LFG, are not primitives in CIG, but rather are names for constituents in particularly common constructions. For example, the *subject* role is simply the first constituent in the SUBJECT-

PREDICATE construction. This constituent is significant because the SUBJECT-PREDICATE is very common, and is included in a number of other large constructions.

CIG constructions are annotated with frequency information. Annotating constructions with frequency information derives from some of the earliest traditions in linguistics and natural language understanding. Grammars which include frequency information have been proposed since the beginnings of modern linguistics. Ulvestad (1960) proposed that *verbs* be annotated with probabilities for each subcategorization frame (well before the term subcategorization was defined). The idea was independently reinvented by Ford *et al.* (1982), who proposed that this ranking of subcategorization frames, known as "lexical preference", be used to make phrase-attachment decisions in parsing.

The *stochastic grammar*, created by augmenting *every* rule of the grammar with a conditional probability, is defined in Fu (1974). The use of such grammars in speech understanding is discussed extensively in Bahl *et al.* (1983). Stochastic grammars for context-free parsing are discussed in Fujisaki (1984), Fujisaki *et al.* (1991), and Jelinek & Lafferty (1991).

The definition of the grammatical construction given in the previous section makes the process of representing a particular construction quite different than the process of defining the *immediate constituent* in American Structuralism, or specifying the *constituent analysis* of a sentence to determine the structure of a *phrase-structure rule* in generative grammar. In CIG, a *construction* is only defined if there is some information (syntactic or semantic) which needs to be expressed which is not predictable from its constituents, and a *constituent* is only defined if it is grammatically required for expressing the construction. American Structuralist and Generative models, on the other hand, have generally proposed methodological principles for deciding how a rule or a sentence was broken down into constituents.

American Structuralism saw a number of specific definitions of the immediate constituent — dating as far back as Bloomfield[5] — couched in terms of their search for a descriptive methodology. In general, these attempt to capture the intuition that "The primary criterion of the immediate constituent is the degree in which combinations behave as simple units" Bazell (1952/1966:284). The most well-known of the specific definitions is Harris's (1946) idea of distributional similarity to individual units, with the *substitutability* test. Essentially, the method proceeded by breaking up a construction into constituents by attempting to substitute simple structures for possible constituents — if a substitution of a simple form, say *man*, was substitutable in a construction for a more complex set (like *intense young man*), then the form *intense young man* was probably a constituent. Harris's test was the beginning of the intuition that a constituent might be formed by constraining some general class of forms to appear in a place. Substitutability was a way of expressing this equivalence notion.

The generative model of Chomsky (1965:197) states that in order to argue for a given constituent analysis, one

> would have to show that these analyses are required for some grammatical rule, that the postulated intermediate phrases must receive a semantic interpretation, that they define a phonetic contour, that there are perceptual grounds for the analysis, or something of this sort. . .

[5]The term *constituent* was popularized by Bloomfield, and Percival (1976) traces Bloomfield's use of the concept to Wundt.

Fillmore (1985) has summarized a number of problems with this generative definition. He showed problems with the *required for some grammatical rule* clause, for example, by showing that various grammatical rules gave conflicting definitions of constituency, and also noted problems with the *phonetic contour* clause.

Many other theories of grammar place much less emphasis on constituency. In the functional grammar of Halliday (1985), the constituent is defined with respect to the *text* rather than the construction. Halliday (1985:28-29) only allows as constituents larger elements such as *Noun Groups* which can play a functional role in the broad structure of the text. In dependency grammar (Mel'čuk 1979), structure is expressed by proposing relations between *words* rather than between constituents of rules.

3.3 The Representation Language

This section will present a simple construction, and outline some representational machinery. The representation we use employs a graphic tree-like notation, where each of the various types of links and positions of elements has a particular significance. The implementation of the interpreter and the grammar uses a Lisp-like sentential form which will appear in certain traces, but in general most of the output from the program will be translated into the more perspicuous graphical notation.

Figure 3.1 shows an example of the CREATE construction. This is the lexical construction which accounts for the verb *create*, and includes the following facts: There is a construction whose name is CREATE, which has a frequency of **177**, and which is a subtype of the VERB construction. The *constitute* of the construction is a semantic structure, which builds an instance of the **Creation-Action** concept. This concept has two subordinate relations, **Creator** and **Created**. Both of these relations are currently unfilled, which is to say they are filled by unbound variables **$a** and **$b**. This construction only has one *constituent*, which includes only phonological (or graphemic) conditions, specifying the form "create".

This representation makes use of a small metalanguage. Construction names appear in bold italic font, like the name *Create* in Figure 3.1. Weak construction names appear in regular italic, like the name *Verb* in Figure 3.1. The bold line between the semantic structure and the word "create" is used to indicate constituency. Constructions with more than one constituent will be discussed in §3.4. The dotted line between *Create* and *Verb* is used to indicate an abstraction link between a construction and a more abstract *weak construction* (see §3.7).

Phonological constraints are represented orthographically, where the double-quotes (") are used to delimit an orthographic constraint. Phonological or morphological representations of a more realistic nature are not discussed.

The semantic language is defined in §3.8, but we summarize its representational features here. The operator **a** indicates an individual of the concept which follows, while the predicates which follow are frame-like role-fillers of this concept, with the filler of each role appearing following the name of the role. The dollar-sign ($) is used to mark variables.

The frequency numbers which appear in the construction are the number of times it occurred per million in the Brown Corpus. The frequency information used in this dissertation comes from two sources, Francis & Kučera (1982) and Ellegård (1978). The former provides frequencies

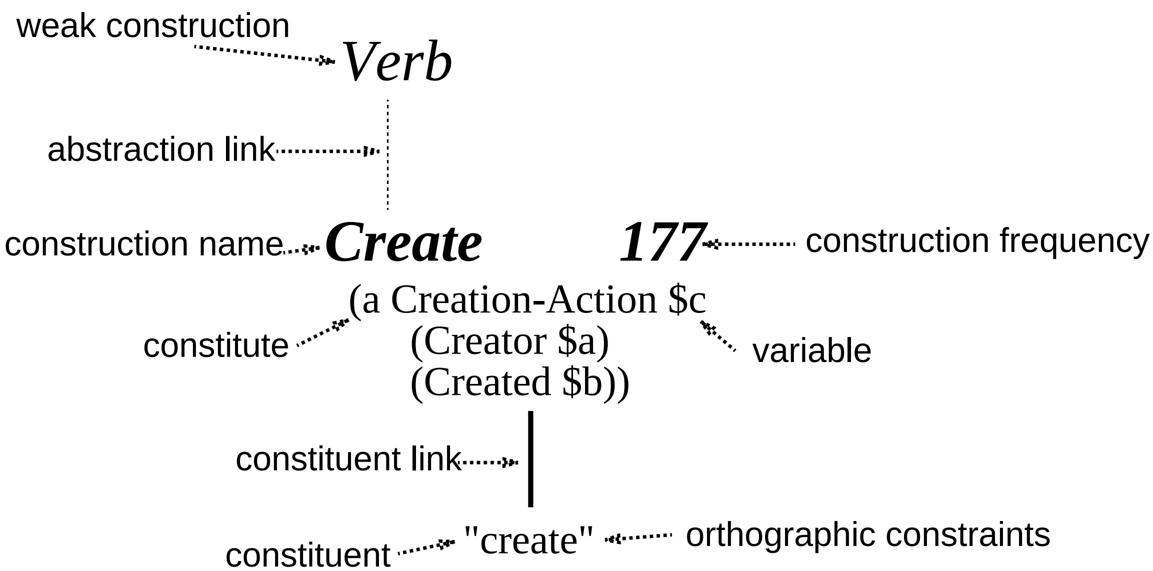


Figure 3.1: The “Create” Construction

for lexical entries and a number of grammatical constructions from the Brown Corpus. In many cases this suffices to assign frequencies to the constructions in our small grammar. Thus, for example, Francis & Kučera lists the CREATE construction as occurring 177 times per million words. Ellegård (1978) provides some syntactic frequencies, also from the Brown Corpus, which are used for some constructions. In many cases, it is not possible to find the exact frequency of a construction, although in these cases it is generally possible to set an upper-bound on the frequency, which is done by using the *less-than* symbol before the frequency. West (1953) was also used to check the frequencies of usage for different senses of the same word.

We conclude with a brief summary and definition of terms of the major elements in a construction.

constitute The semantics associated with a construction. The constitute is the “semantic rule” which is built into a construction. If the construction has constituents, the constitute specifies how the information from the constituents is combined to build the semantics for the construction. The constitute for the CREATE construction is the semantic element **(a Creation-Action $c (Creator $a) (Created $b))**.

constituent A subordinate part of the construction, defined by some informational elements.

unordered constituents Constituents of a construction which does not placed ordering restrictions on its constituents. Unordered constructions are discussed in §3.5.

strong construction A standard construction like CREATE.

weak construction An abstract construction like *Verb* which is used to form an equivalence-class of strong constructions for representation and access. See §3.7 for details.

frequency The number of times a construction appeared in the Brown Corpus per million words.

"word" An orthographic constraint on a constituent, requiring that it match the sequence of letters 'w' 'o' 'r' 'd'. The orthographic constraint from the CREATE construction requires the input to contain the letters 'c' 'r' 'e' 'a' 't' 'e'.

"word" If a constituent has an orthographic constraint like "word", after the interpreter has found "word" in the input and filled in the constituent, the constituent is displayed in italic. Examples of filled-in constituents occur in chapters like Chapter 4 where construction interpretation is discussed.

Name The name of a strong construction, in this case *Create*.

Name The name of a weak construction, in this case *Verb*.

$x A variable named 'x'.

$/x A slashed variable. Slashed variables are used by the integration operation of Chapter 6 to determine how to perform integrations.

$*x A *marked*, or *filled* variable named 'x'. See §6.4.3.

(a . . . **)** The operator **a** indicates an assertion. For example, the element **(a Scale $s)** asserts an instance of the **Scale** concept which is bound to the variable **$s**. See §3.8.

3.4 Constructions with Complex Constituents

In extending the traditional idea of a form-meaning pairing to structures which are more complex than lexical items, the relation between form and meaning also becomes more complex. In particular, the grammatical construction allows a *collection* of forms to have a signification as a group. This notion is called *constituency*, and is based on the familiar use of constituency in American structuralist and generative theories. The use of constituency in CIG is an extension of this traditional sense.

> **The Constituency Principle:** A construction may be composed of constituents, each of which may be defined by any informational predicate, and each of which may be linked to multiple constructions.

The Constituency Principle allows a constituent to be defined by any sort of informational predicate that the representation language allows. There are three classes of these constraints in the system: orthographic, constructional, and semantic. In the remainder of this section, I will discuss each of these types of constraints.

3.4.1 Orthographic Constraints

The orthographic constraints were described in Figure 3.1 in §3.3 above, and are used for lexical constructions. They may also be use for some of the constituents of grammatical idioms that include orthographic/phonological constraints, such as the second and third constituents of "kick the bucket" or "bury the hatchet". As was noted earlier, phonological representations are not addressed here, and this level of knowledge is limited to this simple orthographic representation, in which the orthographic form is simply enclosed in double quotation marks.

3.4.2 Constructional Constraints

The next type of constraint a construction may put on a constituent is to require that it be an instance of a particular construction type. Thus for example the DETERMINATION construction of Figure 3.2 below constrains its first constituent to be an instance of the DETERMINER weak construction. (The second constituent is constrained semantically, while the interpretation for the construction itself is defined by the integration operation I. See Chapter 6 and §3.8 for details).

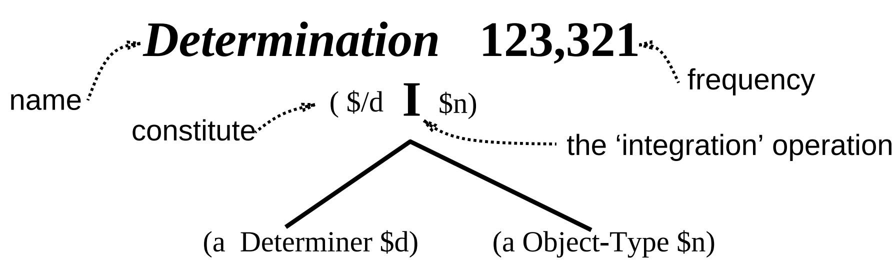


Figure 3.2: The "Determination" Construction

This class of constraints is an extension to the syntactic category constraints that were placed upon constituents in traditional phrase-structure rules. Recall that the only constraint a phrase-structure rule could place on a constituent was that its category be a member of the set of syntactic categories which made up the "universal and rather limited" (Chomsky 1965) non-terminal vocabulary of the grammar. Constraining a constituent to be an instance of a particular construction is more general in two ways than constraining it to have a specific syntactic category. First, rather than a universal and limited vocabulary, the set of allowable constraints is precisely the set of grammatical constructions, which is large and presumably language-specific. In such a system categories like *N* or *A* are not universal formal symbols, but rather names of very abstract weak constructions NOUN and VERB. Second, following directly from this point, because a construction is a pairing of meaning and form, these are not *syntactic* constraints, but rather *grammatical* ones, thus including semantic knowledge as well.

§6.4.2 will show how the *integration* operation uses constructional constraints in combining constructions.

3.4.3 Semantic Constraints

The final class of constraints is semantic specifications. One of the major distinguishing features of this grammar is the ability to define constituents of constructions semantically as well as syntactically. In writing a grammar to describe the constructions of a language, the linguist has traditionally used only syntactic tools. Semantic features were considered generally non-essential to the description of a construction, and to the notion of grammaticality. But a grammatical construction differs from a phrase-structure rule (or similar characterizations of syntactic knowledge) precisely in being a pairing of meaning and form, an abstraction over instances of sets of features, both syntactic and semantic ones. Because these abstractions can be characterized in semantic as well as syntactic forms, semantic information can be used to define constituents of a construction.[6]

Frequently a representational choice is simplified greatly by this ability to use semantic knowledge. But more significantly, especially in exploring non-core constructions of the grammar, semantic features become *necessary* for the minimal description of grammaticality of certain constructions. A theory of grammar which does not allow semantic constraints on constituents of constructions could not adequately represent these constructions.

Since these semantic constraints on a constituent are part of the construction's *definition*, if an instance of a construction violates either syntactic or semantic constraints on its constituents it is as unacceptable as a syntactically ungrammatical sentence in a traditional syntactically-based theory. We use the term *uninterpretable* rather than *ungrammatical* for such constructions, following Stucky (1987). An input to the interpreter is thus *interpretable* if it meets both the syntactic and semantic constraints on some construction, and can be assigned an interpretation by the interpretation mechanism. This distinguishes constructions from phrase-structure rules as well as from the rule-pairs of Montague Grammar, whose semantic rules play no role in grammaticality.

As an example of a construction which requires semantic constraints, consider the HOW-SCALE construction first defined in Jurafsky (1990), which occurs in examples like the following[7]:

(3.1) a. **How old** are you, cook? 'Bout ninety, dey say, he gloomily muttered.
b. **How accurate** is her prophecy?

(3.2) a. **How much** wood could a woodchuck chuck?
b. **How often** does the squire fall off of Dapple?
c. **How quickly** did she finish her work?

(3.3) **How many barrels** will thy vengeance yield thee even if thou gettest it, Captain Ahab?

[6]It is interesting to recall that although American Structuralist models of grammar did not express a formal semantic component, it was considered quite normal to *define* constituents in semantic terms, although they were never represented as such. For example, Huck & Ojeda (1987:1) formalize Bloomfield's (1933) method of defining a constituent as follows:

> If a phonetic string C receives a constant semantic interpretation in sentences $S_1, S_2, \ldots, S_n$, C is a constituent of S_i, $0 \leq i \leq n$.

[7](3.1a) and (3.3) are from *Moby Dick*.

The HOW-SCALE construction has two constituents. The first constituent is the lexical item "how". The second may be an adjective, such as "old" or "accurate" in (3.1a), an adverb such as "quickly" or "often" in (3.2a), or even a quantifier like "many". Specifying this constituent syntactically would require a very unnatural disjunction of adverbs, quantifiers, and adjectives. Furthermore, such a disjunctive category is insufficient to capture the constraints on this constituent. For example, not every adverb or adjective may serve as the second constituent in the construction. Note the uninterpretability of the fragments in (3.4), which have respectively an adverb, an adjective, and a quantifier as their second constituent.

(3.4) a. *How abroad . . . ?

b. *How infinite . . . ?

c. *How three . . . ?

The commonality among the grammatical uses of the construction can only be expressed semantically: the semantics of the second constituent must be *scalar*. A *scale* is a semantic primitive in the representational system, and is used to define traditional scalar notions like *size* or *amount* or *weight*. Note that in (3.4)–(3.4b above all the elements which are allowable as second constituents for the HOW-SCALE construction have semantic components which are scales. Terms like "wide", "strong", and "accurate" meet the traditional linguistic tests for scalar elements (such as co-occurrence with scalar adverbs like "very", "somewhat", "rather", and "slightly"). The elements in the ungrammatical examples (3.4) do not have any sort of scalar semantics. The second constituent of the HOW-SCALE construction may be an adjective, an adverb, or a quantifier so long as it has the proper semantics.

A theory which could not use semantic information to constrain a constituent would be unable to represent the HOW-SCALE construction completely. This includes theories such as HPSG (Pollard & Sag 1987), which assigns semantics to *lexical* constructions but not *syntactic* constructions, as well as (most generative) theories which do not allow semantic information to play a role in the grammaticality of a construction. Figure 3.3 presents a sketch of the representation of the HOW-SCALE construction.

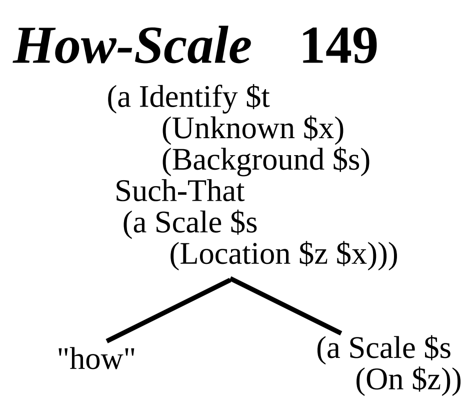


Figure 3.3: The How-Scale Construction

The use of semantic constraints on constituents has another advantage, which is that it enables the grammar to capture generalizations which would not be possible in a syntactically specified grammar. For example, it is sometimes possible to express a construction's constituent *semantically* rather than expressing it as a complex phrase-structure tree, as is possible in TAG (Joshi 1985).[8]

3.5 Constructions with Unordered Constituents

The grammatical construction was defined as an abstraction over linguistic elements. It was noted in Section 4.2.4 that these abstractions could take the form of any collection of linguistic constraints available to the representation language. In this section I discuss *unordered constructions*, constructions which abstract away from the ordering relations that were present in the more specified linguistic elements which they abstract over. This section will not discuss *lexical* constructions which abstract away from ordering relations by maintaining unordered *valence* positions. *Valence-bearing constructions* are lexical constructions which act as functors and specify constraints on their possible arguments. Valence will be discussed in §3.8.

The most common unordered construction in our sample grammar of English is the SUBJECT-PREDICATE construction. This construction expresses the relation between the elements *"he"* and *"was"* in the sentences of (3.5a) and (3.5b):

(3.5) a. He was playing a very funky beat.

b. Was he playing a funky beat?

This definition of the relation between subject and predicate is different from the traditional one, which relates the subject with the entire verb-phrase. Here, the relation is solely between the subject and the head verb "was", and specifies subject-verb agreement and the semantic relation between the subject and the verb.

Figure 3.4 shows the representation of the ACTIVE-SUBJECT-PREDICATE construction, which is a subtype of the SUBJECT-PREDICATE construction for active clauses. The construction contains two constituents, one constrained to be a VERB, and the other one constrained to be a **Actor** (see the next paragraph) and to integrate into the valence structure of the verb (the large symbol **I** is used to indicate the integration operation; see §6.4.3). The *arc* drawn between the two constituents indicates that they are *unordered*.

The construction builds its semantics by integrating its two constituents, **$v** and **$s**. The VERB constituent in the integration has been marked with a slash (**$v**). This indicates that the verb will serve as the matrix for the complement. The other constituent of the construction, labeled **$s**, has been constrained to fill the **Actor** role (Foley & van Valin 1984), which abstracts over those thematic roles which generally act as grammatical subjects. This will constrain which valence

[8]Paul Kay and Charles Fillmore (personal communication) have noted a number of other examples in which constructions must include semantic constraints, such as the constraints on the complements of the verb *feel*, as well as in the constructions in (i):

(i) He attended to her every thought/wish/*washing machine.

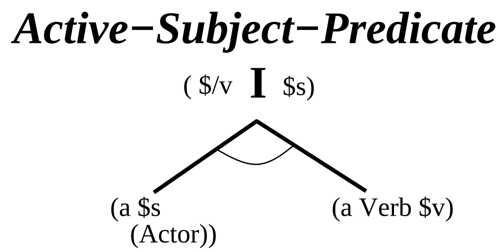


Figure 3.4: The Active-Subject-Predicate Construction (omitting agreement)

role of the verb the first constituent can fill, and is used by the integration operation (see §6.4.3). More details on valence semantics are in §3.8.

Note that unordered constructions are different than the unordered rules of Boas (1975) or the Immediate Dominance rules of Gazdar *et al.* (1985) because unordered constructions in CIG are the exception rather than the rule; most constructions include ordering specifications. CIG *allows* abstraction over ordering, but does not require it.[9]

3.6 Constructions with Linked Constituents

A grammatical construction may constrain a number of its constituents to act as co-constituents in an instance of some other construction. This may be viewed as an extension of the method described in §3.4 of constraining a single constituent of one construction to be an instance of some other construction. Multiple constituents which are constrained to appear in multiple constructions are called *linked constituents*.

Consider as an example the WH-NON-SUBJECT-QUESTION construction (Jurafsky (1990)). This construction accounts for sentences which begin with certain wh-clauses, where these clauses do not function as the subject of the sentence. Examples include:

(3.6) **How** can I create disk space?

(3.7) **What** did she write?

(3.8) **Which book** did he buy?

The construction has four constituents. The first, indicated in bold type in the examples above, is a wh-element. The second is an auxiliary verb, and participates together with the third constituent in the SUBJECT-PREDICATE construction. The second and fourth constituents are

[9]Of course languages may differ in the amount they make use of ordered versus unordered constituents, and the relative importance of *valence arguments* versus *constituents* in the grammar. For example in representing non-configurational languages like Warlpiri (Hale 1983), a construction grammar would make greater use of lexical valence structures and semantic constraints on unordered constituents, and less use of ordered constituents. This use of non-constituent means of capturing generalizations is similar to the LFG approach to Warlpiri (Simpson & Bresnan 1983), which allows generalizations to be captured at the level of f-structure, without the assumption of a non-configurational parameter.

constrained to occur in an instance of the VERB-PHRASE construction. The representation for the construction appears in Figure 3.5 below.

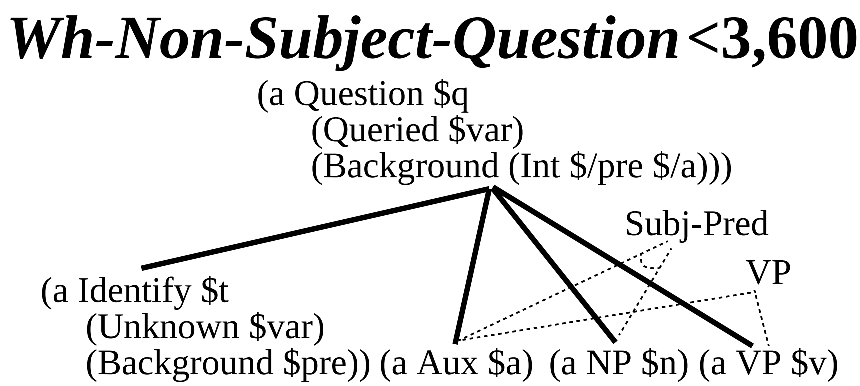


Figure 3.5: The Wh-Non-Subject-Question Construction

Consider the definitions of each of the constituents in this construction. The first constituent is defined by the **Identify** concept. The **Identify** concept characterizes the *wh-* constructions — it instantiates a frame in which the identity of some element is in question, and where some background information is provided to help identify the element. For example if the *wh-* element is the lexical item “why”, the unknown element is the reason or cause for some action, and the background information would specify the action itself.

While the first constituent of the construction is a simple one, the second, third, and fourth constituents are all linked constituents, in that they each participate in multiple constructions. First, note that the second and third constituents are related by the SUBJECT-PREDICATE construction. This is necessary both to enforce the agreement relation between these constituents, and to ensure that the semantics of the auxiliary and the semantics of the subject noun-phrase are combined in the correct way. For example, the structure of sentence (3.6) appears in Figure 3.6. Note that the SUBJECT-PREDICATE construction must hold between the words *I* and *can*.

Similarly, the second and fourth constituents are related by the VERB-PHRASE construction. This is necessary to insure that any constraints that the auxiliary places on its complements (such as requiring that their inflectional category be **bare-stem**) are placed on this fourth constituent.

The WH-NON-SUBJECT-QUESTION illustrates the difference between Construction Grammar approaches to syntax and more traditional generative grammar approaches. For example, the fact that the AUX element precedes the NP was handled in transformational grammar with the *Aux-Inversion* transformation, which derived these structures from ones with “canonical” SVO ordering. CIG is non-derivational, and so the facts about ordering of individual constructions are represented locally in each construction. More recent theories, like HPSG, represent ordering information with general principles about the linear precedence of heads and complements; presumably a theory like HPSG could be extended to allow some ordering information to be construction-specific.

It is interesting to note that allowing an element to simultaneously instantiate multiple constructions was discussed as early as Wells (1947:95), who remarks that “Our definition of the term

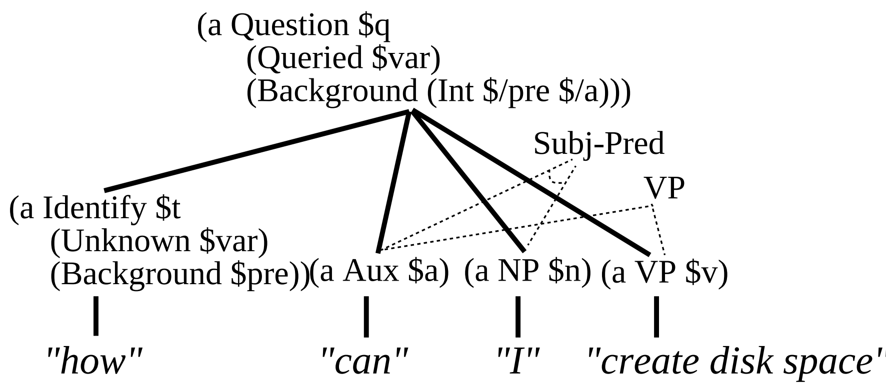


Figure 3.6: The structure of “How can I create disk space?”

construction allows an occurrence to belong to more than one construction. . . ”. See Fillmore (1989a) for further discussion of multiple instantiation.

Note that a parse which allows complex constituents does not have a *parse tree*, but a parse *graph*. Recall that a *tree* is defined as a directed acyclic graph (DAG) in which exactly one vertex, the *root*, has no entering edges, and in which every other vertex has exactly one entering edge (Aho *et al.* 1974). Because a complex constituent can belong to more than one construction, these constraints do not hold. Note, for example, that in the WH-NON-SUBJECT-QUESTION construction in Figure 3.5, the last three constituents all have more than one entering edge. (See Karlgren (1976) for an earlier proposal to extend the notion of ‘parse tree’ to ‘parse graph’.) This parse graph complexity is less important for CIG than it is for other theories (such as the autolexical syntax of Sadock (1987), where it is an important theoretical distinction) since CIG is embedded in a model of sentence *interpretation*. As such although the syntactic structure of individual constructions is significant, the theory makes no claims about the syntactic structure of sentences, but only about their interpretations.

3.7 Weak and Strong Constructions

Construction grammars like CIG make no theoretical distinction between the lexicon and the rest of the grammar, following the *Uniformity Principle*. But the grammar-lexicon distinction proved useful to earlier theories of grammar in distinguishing *productive*, semantically-coherent rules or processes, from lexicalized, semi-productive, more idiomatic rules or processes. Until now, the constructions we have described correspond to the productive, semantically coherent rules (although of course without the derivational algorithms that characterize rules in the derivational-generative model). It makes no sense to speak of the “productivity” of a construction — constructions are always completely productive, in the sense that they can be employed whenever the conditions on their constituents and meaning are met.

How, then, are we to describe the more “lexicalized”, “sporadic” rules? They cannot be regular grammatical constructions, because they do not have a coherent semantics, and because they allow exceptions. Recall that these rules were proposed in generative grammar to capture *generalizations* across elements — for example the abstract similarity between verbs and derived nominals. Following this intuition, we propose that these non-productive rules can be represented as a sort of *abstract construction,* which augment the representation of standard constructions by abstracting over them in an *abstraction hierarchy*. We call these new constructions *weak constructions,* and will refer to the standard constructions as *strong constructions*. Strong constructions are defined *intensionally*, by specifying constraints on the constitute and the constituents, but weak constructions are defined *extensionally*, by specifying the set of constructions they abstract over. Weak constructions are defined as follows:

> **The Weak Construction:** A weak construction w is defined by specifying a construction name n, a constitute c, and a set s of strong and/or weak constructions over which w abstracts,

Weak constructions resemble the *subregularities* which Wilensky (1990) has proposed for the representation of lexical semantics and the *broad-range* rules which Pinker (1989) has proposed for representing lexical argument-structure generalizations. Weak constructions are used in the grammar for two purposes. First, they serve to structure the grammar by linking together strong constructions in a way that is useful for access, for creating new constructions, and for learning (see Vennemann (1974) for similar arguments). Second, having weak constructions allows the grammar to specify an equivalence-class of constructions which can be used to constrain the constituents of other constructions.

Consider an example of this use of a weak construction to constrain a constituent. A significant fact about CIG is that it defines *lexical categories* as weak constructions, rather than as their traditional role of representational primitives (§3.7.1 will discuss the distinction in further depth). Thus traditional lexical categories such as AUX, or VERB are represented as weak constructions, and thus act as an equivalence class for certain strong constructions. For example, the weak construction AUX abstracts over the strong constructions CAN, WILL, and other auxiliaries. This enables other constructions to constrain their constituents to be AUXES, without having to define a construction for each individual auxiliary, which would cause a huge proliferation of constructions. This was the case in the WH-NON-SUBJECT-QUESTION construction of Figure 3.5 above, which constrained its second constituent to be an AUX.

Figure 3.7 shows a number of examples of weak constructions. These include the lexical weak construction NOUN, the morphological weak (nominalization) construction -EE, and the weak construction VP.

Note that in each of these cases, weak constructions are at non-terminal positions of the abstraction hierarchy, while the strong constructions are at the terminal nodes. The weak constructions express abstractions over the strong constructions which are below them in the hierarchy. This hierarchy is learned in the same way that constructions are learned, by successively forming abstractions over linguistic instances. When strong constructions are learned in this way, the particular instances which gave rise to the construction are not stored in the grammar — only the generalization, the construction, remains. But when weak constructions are formed by abstracting over a set s of strong constructions, the instances which gave rise to the weak constructions (i.e., the set s of strong constructions) remain in the grammar.

The next three sections will consider examples of strong and weak constructions. §3.7.1

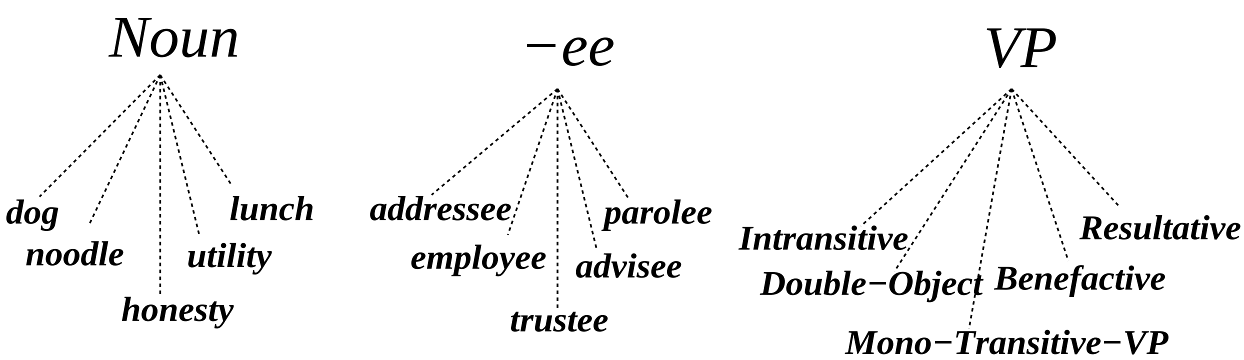


Figure 3.7: Weak Constructions

shows how traditional *lexical categories* are modeled in CIG as weak constructions. §3.7.2 discusses larger weak construction like the VP construction, and §3.7.3 shows how the weak/strong distinction can be used in morphology to capture the notion of *productive* vs *non-productive* rules and *inflection* vs *derivation*. Finally, §3.7.4 places the notion of weak construction in a historical context by summarizing previous models of abstraction in linguistic knowledge, such as the *redundancy rule* or the *metarule*.

3.7.1 Lexical Weak Constructions

The idea of *weak construction* can be applied to *lexical categories*. In generative grammar, for example, lexical categories are representational primitives. Early generative grammar included *lexical insertion rules* which mapped lexical categories into individual lexical items. Modern versions of generative grammar include a small number of primitive lexical categories, and mark each lexical item with a particular category.

CIG does not assume that the grammar contains a small number of representational primitives (such as the *N*, *V A*, and *P* of Chomsky (1986), or the nine lexical categories defined by Jackendoff (1977)). Instead, lexical constructions like NOUN and VERB are *weak constructions*. Representing these as weak constructions instead of primitive lexical categories means first that these are not necessarily limited to a very small number, and second that as constructions they include a *semantic* component. Because they are weak constructions, they are defined extensionally, and do not have a completely consistent semantics. But they are used by other constructions to specify equivalence classes of constructions. For example, the beginning of this section noted that the weak construction AUX abstracts over the strong constructions CAN, WILL, and other auxiliaries. This enables other constructions (such as the TAG-QUESTION construction, for example) to constrain their constituents to be AUXES, without having to define a TAG-QUESTION construction for each individual auxiliary, which would cause a huge proliferation of constructions.

Lexical categories are particularly significant weak constructions because they abstract over so many individuals and because of their relatively general semantic correspondence. Of course it has been observed since Aristotle that a rough correspondence can be drawn between lexical

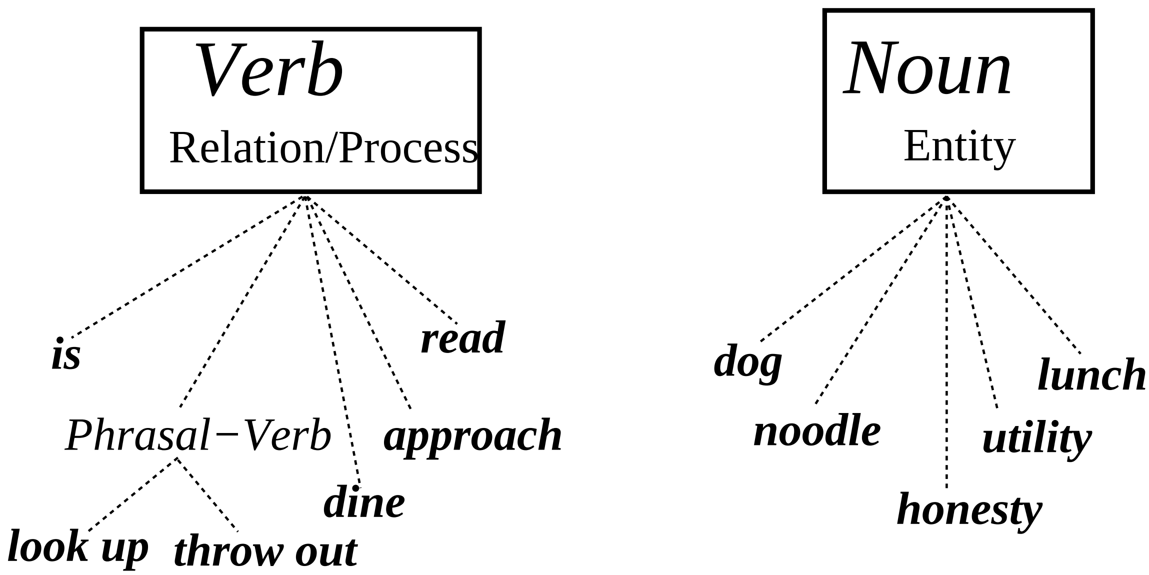


Figure 3.8: Lexical Categories as Weak Constructions

categories and an ontological partitioning of the world. The correspondence is rough because of its numerous exceptions. For example while nouns generally refer to objects and verbs to actions, deverbal nominalizations such as *destruction* seem to be more like verbal actions than nominal objects.

But the many exceptions to this analysis caused the real semantic nature of lexical categories to receive less attention than it deserved. As Miller & Johnson-Laird (1976:527) note, "perhaps [traditional grammarians] did not really mean that everything labeled by a noun is a concrete object; perhaps they meant that when you use a noun to label something, you tend to conceptualize it as if it were a concrete object". Thus lexical categories might have a very abstract semantics like **Entity** or **Process/Relation** (see Langacker (1987)). Because lexical weak constructions are so high in the abstraction hierarchy, they become something like what Lakoff (1987) called "central principles" or Chomsky's (1986) "Canonical Structural Realizations" of semantic concepts.

Representing lexical categories as weak constructions captures the intuition that lexical categories are an *abstraction* over lexical entries with an associated *abstract semantics*, while using a mechanism, the weak construction, which is independently motivated in the grammar. Thus this aspect of CIG is more elegant than the traditional generative practice, which required explicitly *specifying* a distinct and primitive set of categories.

Versions of construction grammar which include the *lexical network theory* of Norvig & Lakoff (1987) and Brugman & Lakoff (1988) allow even richer relations among lexical constructions than the abstraction relations characterized by weak constructions. Although such relations are not discussed in this dissertation, presumably CIG could be augmented with these networks, particularly in those cases in which the networks are shown to be used in language processing.

3.7.2 Larger Weak Constructions

Just as weak lexical constructions abstract over lexical constructions, larger weak constructions abstract over larger constructions. One particularly common weak construction is the VP or VERB-PHRASE construction, which abstracts over a number of strong constructions which relate a verb to its possible complements. Figure 3.9 shows the weak VP construction and a number of the constructions that it abstracts over.

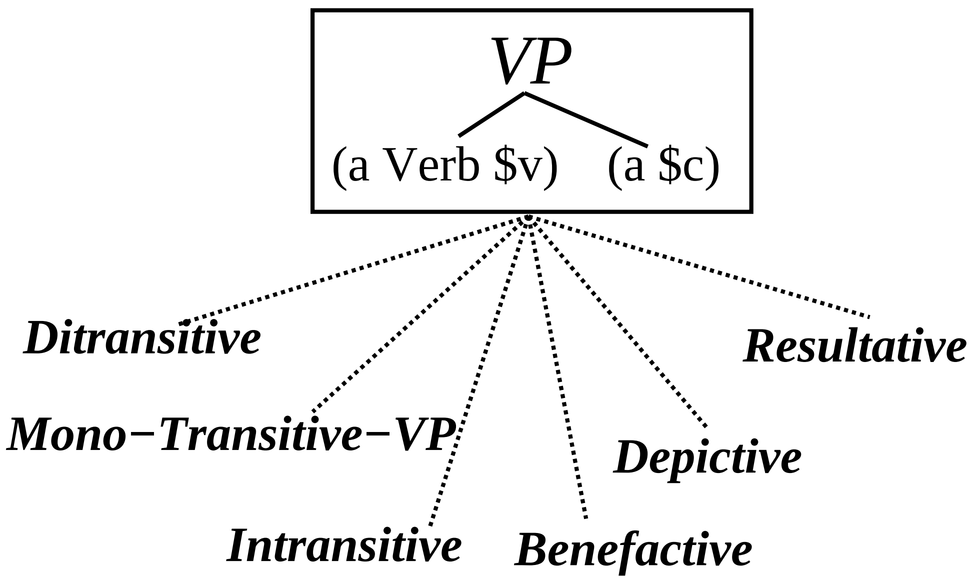


Figure 3.9: The Weak VP Construction

Each of these strong verb-phrase constructions specifies a kind of verb and the kind of complements the verb can take. In general these constraints are expressed semantically. §7.6.2 discusses the BENEFACTIVE construction, while the DITRANSITIVE construction is discussed in Goldberg (1989).

The use of the weak construction is one way to capture relationships among a number of constructions. Many versions of construction grammar have proposed more powerful ways to capture such relations, such as the lexical networks discussed in the previous section. For larger constructions, the cognitive grammar proposals of Lakoff (1984) and (1987) show how a number of *there*-constructions, such as the CENTRAL DEICTIC construction and the PERIPHERAL DEICTIC construction, can be structured by relating the constructions with the *Based-On* relation. The details of the analysis and the definition of each of these construction is specified in *Case Study 3* of Lakoff (1987), and in Lakoff (1984).

3.7.3 Morphological Weak Constructions

"I never heard of 'Uglification,"' Alice ventured to say. "What is it?"

The Gryphon lifted up both its paws in surprise. "Never heard of uglifying!" it exclaimed. "You know what to beautify is, I suppose?"

"Yes," said Alice doubtfully; "it means — to — make — anything — prettier."

"Well, then," the Gryphon went on, "if you don't know what to uglify is, you are *a simpleton,"*

— Lewis Carroll, Alice's Adventures in Wonderland

The distinction between strong and weak constructions corresponds quite naturally to the distinction in traditional morphology between *inflection* and *derivation*, and in classic generative grammar between *productive rules* which were assigned to syntax, and *non-productive rules*, assigned to the lexicon.

Consider, for example, the traditional account of the English rules for nominalization. According to this account, English has two classes of nominalizing rules, the productive action or gerundive nominals, and the non-productive "derived" nominals. In such a theory the verb "destroy" has two lexical nominalizations — the productive or gerundive "destroying", and the derived or non-productive "destruction". The nominal use of the gerundive "destroying" is productive in the sense that if a new verb entered the language, say "to xerox", the native speaker would automatically be able to speak about "xeroxing", and in the sense that the semantics of the new word *"xeroxing"* would be predictable. The rule which derives the form "destruction", however, is non-productive. For each non-productive nominalizing suffix (like "–ity", "–ure", etc.) it is necessary to enumerate some or all of the lexical items which undergo the rule (see Aronoff (1976)), and specify the semantics of the combination.

By allowing both weak and strong constructions, CIG can represent both the productive and non-productive nominalizations. Productive nominalizations such as the English gerundive are represented as strong constructions. An important early argument for nominalization rules was that they capture the generalization between the argument structure of verbs and nominalizations. In CIG the correspondence between the argument structure of verbs and the argument structure of the gerundive is captured because the lexical verb and the suffix combine to form an instance of the gerundive construction. Thus the gerundive construction maintains the same argument structure as the verb. Figure 3.10 shows an example of a strong (the gerundive) and a weak (the -EE) nominalization construction.

The various non-productive nominalizations are represented as weak constructions. This means that they are abstractions over individual instances of nominalizations. A characteristic feature of the non-productive nominalizations is that their semantics is rarely fully predictable from the semantics of the underlying verb — they tend to differ in idiosyncratic ways. Diachronically speaking, the non-productive nominalizations have undergone semantic drift. But note that this semantic drift will generally not carry the nouns far enough to change their thematic structure. Thus the similarity in the argument structure between non-productive nominalizations and verbs is not a syntactic fact, as it is in the traditional model, but a semantic one.

There is an extensive psycholinguistic literature on weak and strong morphological constructions, generally phrased in terms of the distinction between inflection and derivation. For example, there are a number of results supporting the idea that inflection is represented as a distinct construction, and that the lexicon does not compile out an inflected form of each entry. Butterworth (1983) has called the latter the Full Listing Hypothesis. As Cutler (1983) noted,

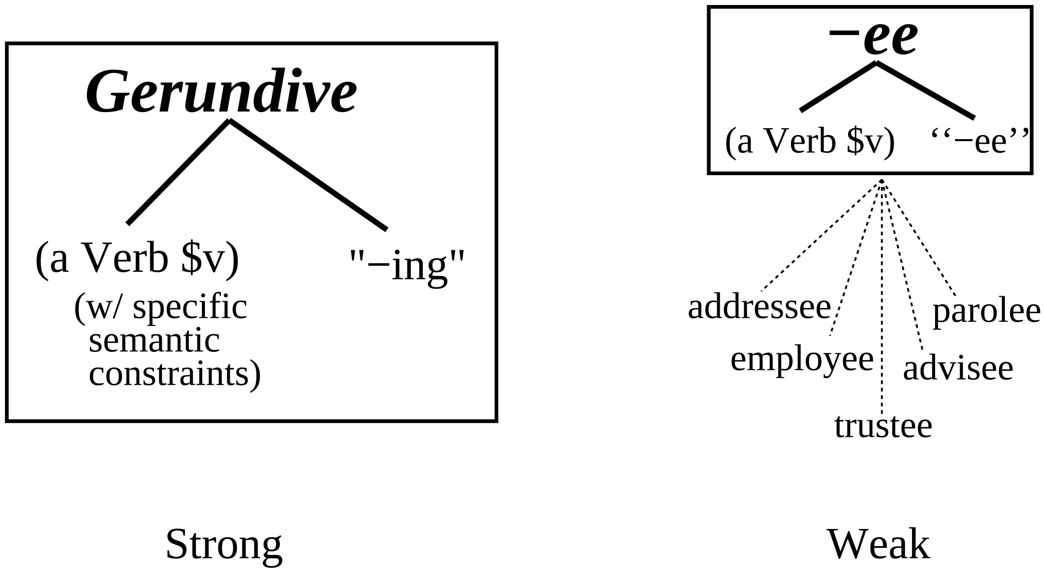


Figure 3.10: Strong and Weak Nominalization Constructions

"There is abundant evidence that words inflected for tense or number do not have lexical representation independent of their base form, and that base word and inflection are separated in language processing". Cutler gives a number of references besides the Stanners *et al.* (1979) study mentioned below. In a more linguistic vein, Hankamer (1989) has made complexity arguments from Turkish morphology, showing that no version of the Full Listing Hypothesis is possible, at least for agglutinative languages like Turkish. Hankamer showed that the number of forms which can be created from a single verbal or nominal entry in Turkish is larger than could fit in the mental lexicon.

In addition to evidence for morphological strong constructions, there is extensive psycholinguistic evidence that *inflected* and *derived* forms are represented differently. For example, Stanners *et al.* (1979) showed that accessing the inflected forms of verbs (such as the past tense form, or the gerund) caused the root form of the verb to be primed very strongly. This would be expected if the past-tense inflection is represented as a separate strong morphological construction, because the past tense form would be created as a combination of the root-verb construction and the inflection construction. Stanners *et al.* found, however, that accessing *irregular* past tense verbs only weakly prime the root verb. Again, this would be expected if irregular or derivational suffixes are represented as weak constructions, since the past tense form of the verb is *not* created by combining any of the weak past-tense constructions with the root form of the verb. Rather, the irregular past tense forms are listed explicitly in the lexicon.

The fact that some weak priming of base forms does occur led Stanners *et al.* to suggest that the derived forms are represented in some way that relates them, albeit weakly, to a base form. The weak morphological construction may play this role. A number of other studies which show distinct but linked representation are summarized by Cutler (1983). An alternative view, in which the weak link between the derived and base forms is captured by some sort of analogical process, is presented in Derwing & Skousen (1989). Sadock (1984) and Wilensky (1990) also note the need to include such local generalizations in the lexicon.

3.7.4 Related Models of Abstraction

The idea of using weak constructions as an abstraction over other constructions derives from two traditions in generative linguistics. The first is the tradition in linguistics of proposing mechanisms to capture generalizations across sentences and rules, such as Chomsky's early arguments against context-free grammar. Chomsky did not try to show that context-free grammar was unable to account for the syntactic facts of English. Instead, he showed that given certain assumptions, context-free grammar was unable to capture certain interesting *generalizations* across sentences of the language. His formalization of the theory of grammar employed structural transformations as a device for capturing such generalizations. During the same period, Halle (1959) proposed the use of *redundancy rules* to express phonological generalizations. This use of redundancy rules was extended to generalize over lexical entries (see Jackendoff 1975 and Bresnan 1978).

The second major tradition is the idea of capturing the distinction between productive and non-productive rules by locating non-productive rules in the lexicon. This idea was suggested by Zimmer (1964), drawing on the traditional notion (see Bloomfield 1933, for example) of the lexicon as the repository of all arbitrary terms:

> The implication of such a model for the linguistic behavior of speakers of English is that a number of forms such as *untrue*, *unhappy*, *unkind* are learned as lexical items like *true*, *happy*, *kind* while other forms in *un-* can reasonably be accounted for as the output of productive rules that should be given a place in the equipment we assume the speaker to be operating with. (p. 85)

Zimmer goes on to say that these lexicalized processes which relate forms like *true* and *untrue* still constitute important regularities which must be dealt with:

> We would therefore say . . . that the derivation of nouns in *-ling* should be excluded from a "generative" and dealt with in an "analytic" morphology.

These two traditions were merged in the next theoretical advance in mechanisms for linguistic abstraction — the *Lexicalist Hypothesis* of Chomsky (1970), which proposed generalizing over *rules*, rather than sentences. The lexicalist hypothesis proposed that *productive* nominalizations be accounted for by syntactic rules, but that *non-productive* nominalizations be listed in the lexicon. In order to capture the generalization between the argument structure of the *lexical* nominalizations and the related verb, Chomsky (1970) proposed the X-Bar Convention, the use of generalized cross-categorical phrase-structure rules (see also Jackendoff (1977)). This allowed the generalization between the verb and the derived nominal to be captured by sharing similar phrase-structures, rather than by a transformational rule.

The *metarules* of GPSG (Gazdar 1982 and Gazdar *et al.* 1985) combine elements of each of these mechanisms, which were basically an extension of the *redundancy rule* to generalized over *phrase-structure* rules but applied in a lexical fashion.

More recently, many versions of construction grammar, including Lakoff (1987), Norvig & Lakoff (1987), and Brugman & Lakoff (1988), have proposed capturing generalizations among constructions with network-style relations among constructions.

In parallel with the development of mechanisms for capturing generalizations in generative linguistics, the abstraction hierarchy was proposed in the fields of computational linguistics and knowledge representation, beginning with Quillian (1968) and Collins & Quillian (1969), and continuing with knowledge representation languages such as Fahlman (1979), Bobrow & Winograd (1977), Brachman & Schmolze (1985), Wilensky (1986) and Norvig (1987). The abstraction hierarchy was originally proposed to represent solely semantic knowledge. The idea of using it to explicitly represent *linguistic* knowledge, and hence the correspondence between meaning and form, and to capture generalizations across this correspondence, was first proposed by Bobrow & Webber (1980), quickly followed by a number of other models, including Hudson (1984), Jacobs (1985), Flickinger *et al.* (1985), Pollard & Sag (1987), Jurafsky (1988).

The use of weak constructions to abstract over strong constructions is similar to these last proposals, but differs in not using the notion of *inheritance*. Inheritance is a process by which concepts lower on a hierarchy augmented with information from concepts higher on the hierarchy. Thus concepts which are higher in an inheritance hierarchy abstract over lower concepts, producing a more efficient representation, since redundant information is removed from the lower concepts. Inheritance thus strongly resembles the *redundancy rule* mechanism, or the *metarule* mechanism of GPSG. One aspect of the resemblance is that both inheritance and redundancy-rule/metarule mechanisms could be viewed as *generative* mechanisms or as *redundant* mechanisms. For example, redundancy rules were proposed in two ways (Jackendoff 1975), one in which they merely abstracted over two fully-specified lexical entries (the *full-entry* theory) and one in which they were generatively employed to produce a second lexical entry from a first (the *impoverished-entry* theory). Shieber *et al.* (1983) note that this holds for metarules as well. Similarly, inheritance mechanisms may employ what Fahlman (1979) has called *virtual copies* or *real copies*. In virtual copying, lower structures do not duplicate the information in more abstract structures, and thus inference mechanisms must search up the hierarchy to fully instantiate a concept. This is the version of inheritance which was first defined in Quillian (1968). In real copying, lower structures *do* include all the information from higher structures; they are *compiled out*, and the abstraction captured by the higher structures is not used at run-time.

Weak construction abstraction is very similar in spirit to inheritance or meta/redundancy rules, but differs somewhat in application from both the compiled-out version and the generative version of these mechanisms. First, weak constructions are not used to *generate* new rules; the strong constructions which are abstracted over by weak constructions are completely filled-out. Second, compiling out information from the inheritance hierarchy in advance would violate the Interpretive Hypothesis of Chapter 2, because the weak constructions would then play no role in language processing. Indeed, Shieber *et al.* (1983) conclude that viewing metarules as merely a redundant way to structure the grammar allows them to play no role in processing. However, although they act as redundant information, weak constructions *are* used in processing, for example in aiding the access mechanism, as discussed in the beginning of §3.7 (see Stowe (1984) for a similar suggestion).

3.8 The Representation of Lexical Semantics

This section turns to the simple semantic representation language that is used in CIG. We made recourse in previous sections to an intuitive notion of a semantic representation language. In this chapter we discuss this idea further, and explore issues in the representation of lexical semantics, concentrating on describing the kinds of semantic expectations or conceptual constraints that can be used in processing to help access or integrate other constructions. In general, then, we will express only as much semantics as is necessary to express constraints on constituents and to combine constituents.

3.8.1 The Representation Language

We have chosen a simple frame-like representation language to represent the conceptual domain of grammatical constructions. Most modern representation languages (KRL (Bobrow & Winograd 1977), NETL (Fahlman 1979) KL-ONE (Brachman & Schmolze 1985), FRAIL, KODIAK (Wilensky 1986) and (Norvig 1987), SNEPS (Maida & Shapiro 1982)) have basically proposed to represent semantics by augmenting predicate logic by structuring concepts into *frames* or *schemata*, following the insights of Bartlett and Minsky. The language which is used by CIG has two components; the *definitional language*, which is used to define various concepts and their slots, and the *assertional language*, which is used in each grammatical construction to make assertions about the semantics of the construction and its constituents. Both of these languages consist of a large number of concepts and a small set of operators which allow the concepts to be defined and manipulated. The concepts themselves are the familiar ones that exist in every such semantic language. These include abstract concepts like **Actions**, **Events**, or **Objects**, as well as more specific concepts like **Creation-Action** or **Scale**.

(3.9) shows the format of an *assertion* in the assertional language:

(3.9) `(a Scale $c)`

An assertion like (3.9) consists of at least three elements. These are an *operator*, a *concept*, and a *variable*. The operator for the assertion above is **a**. The operator **a** creates an instance of the concept which follows. Operators are modeled after the *frame determiners* of Hirst (1986), which were used as a metanotation for Frail. The second element is the name of the *concept* to which the operator applies. Thus (3.9) creates an instance of the **Scale** concept. The third argument, the *variable*, is bound to this new concept. The variable in (3.9) is **$c**. Variables in CIG are marked by a dollar-sign ($). Thus the meaning of (3.9) is that the variable **$c** is bound to an instance of the **Scale** concept. Variables in CIG are *logical variables* like the variables of functional or term unification, rather than the *location*- or *content-pointer* variables of standard programming languages (see Pereira & Shieber (1987)).

As in frame-oriented languages like Frail or KL-ONE, concepts are structured entities, with subparts, *slots*, which place constraints on their fillers. We assume the definition of these slots in the definitional language to be the standard one, in which a frame-name creates an implicit $\forall$, and each slot of the frame instantiates an implicit $\exists$ in the scope of the $\forall$. However, we assume that each slot is a predicate, which can take any number of arguments, rather than a single slot-filler. A CIG assertion may refer to the slots of a concept in order to further constrain them, or in order to use the slot information in building construction interpretations. For example, the **Scale** concept is defined in the representation language to have a number of possible slots, which represent such things as *which objects* are on the scale, or the *location* of objects on the scale, or the *domain* of the scale. The slots of the concept can be used in specifying the constitute or a constituent of a construction. For example, the assertion in (3.10) shows the representation of a **Scale** which specifies the **On** predicate. The **On** slot is used to express the relation between a scale and some object on the scale. It takes a single argument, which is filled in (3.10) by the variable **$z**, indicating that whatever is bound to **$z** is *on* the scale **$s**.

(3.10)
```
(a Scale $s
    (On $z))
```

The meaning of (3.10) is thus that there is some instance of a scale **$s** with some object **$z** on the scale. Remember that this assertion is not the *definition* of the scale concept, but rather the semantics associated with a particular constituent which *makes use* of the scale concept.

The assertion in Figure 3.10 is the semantics of the second constituent of the HOW-SCALE construction from Figure 3.3 above, repeated in Figure 3.11. Consider now the semantics of the *constitute* of the HOW-SCALE construction. It begins with the **Identify** concept. The **Identify** concept characterizes the semantics of all the *wh-* constructions. Its meaning is that the identity of some element is in question, with respect to some background information about the element. For example as §3.6 mentioned, if the *wh-* element is the lexical construction *why*, the unknown element is the reason or cause for some action, and the background information would specify the action itself. For the lexical construction *who*, the background information is that there is some person **$x**, while the unknown is the identity of **$x**.

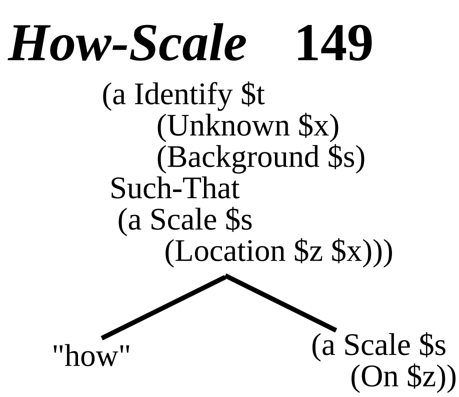


Figure 3.11: The How-Scale Construction (from Figure 3.3)

For the HOW-SCALE construction, note first that the **Unknown** element is filled by the variable **$x**, while the **Background** element is filled by the variable **$s**. Constraints on these variables are specified by including further assertions after the operator **Such-That**. The assertion after the **Such-That** operator in the constitute of Figure 3.11 is repeated in (3.11). It places constraints on both the **Unknown** and **Background** slots:

(3.11)
```
(a Scale $s
   (Location $z $x))
```

The meaning of (3.11) is that there is some instance of a scale **$s** where some object **$z** is located at position **$x** on the scale. That is, the first argument of the **Location** slot is an object on the scale, and the second argument is the *location* of the object on the scale.

Thus the meaning of the entire semantics of the constitute of the HOW-SCALE construction is that there is some **Scale** with some *object* on it, and the *location* of the object on the scale is in question. §6.4.3 shows how the *integration* operation combines the semantics of the constituent with the semantics of the constitute in producing an interpretation for a construction.

See Talmy (1977, 1978, and 1986) for further discussion of the semantics of scales.

3.8.2 Valence

The previous section defined the language which is used to represent the semantics of constructions *constitutes* and *constituents*. It showed that a constitute or constituent can be defined by creating an instance of an assertion and specifying some of its slots.

This section shows how these slots act as the *valence* of a construction, in effect creating expectations for slot fillers. The term *valence* was used originally by Tesnière (1959) to indicate the *number* of arguments a verb might take. Construction grammars like CIG extend the term to mean the number and type of open fillers which are associated with a construction. The *valence* properties of a predicate include such constraints as the *thematic* roles of an argument as well as the *syntactic subcategorization constraints* on an argument.

The *valence theory* thus generalizes over earlier theories of argument specifications, which have used such representations as *subcategorization frames* or *selectional restrictions* (Chomsky 1965), *case frames* (Fillmore 1968 and 1977) or *thematic grids*.[10]

Consider the scale from (3.10) above, repeated as (3.12):

(3.12)
```
(a Scale $s
   (On $z))
```

(3.12) indicates an instance of a scale with some element **$z** on the scale. Because the assertion (3.12) does not specify a filler for the **On** slot, the variable **$z** is an *open variable*. This open variable **$z** is a *valence argument* of the assertion. A *valence argument* is defined as any unfilled slot of an assertion. Thus (3.12) might also be interpreted as "an instance of a scale which has a single valence argument which fills the **On** slot". Since the semantics of scalar adjectives includes the assertion (3.12), this valence corresponds to the valence of scalar adjectives.

While in CIG any construction can have valence, most previous work on valence-like theories has focused on verbs. This is especially true of theories of *thematic roles* and of *subcategorization*. Consider the representation of verbal valence in CIG. The construction CREATE described in §3.3 above has the assertion (3.13) as its constitute:

[10]The idea that slots of concepts in a representation language correspond to grammatical arguments was first proposed by Quillian (1968:244-245). Charniak (1981) discussed the relation between case roles and frame-slots.

(3.13)
```
(a Creation-Action $c
   (Creator $a)
   (Created $b))
```

Example (3.13) asserts an instance of the **Creation-Action** concept, with two slots, **Creator** and **Created**. The fillers of **Creator** and **Creation** are the variables **$a** and **$b** respectively. The fact that these two variables are unbound means that they are *open variables*, and hence they are *open valence arguments* of the CREATE construction.

The CREATE construction only has two valence arguments, but the **Creation-Action** concept has many more slots which are not included in the CREATE construction. For example, **Action** concepts have slots for the **Time** and **Location** of the action. These slots are not listed in the CREATE construction, and so they are not *open valence arguments* of the construction, but they can play a role in the processing of the CREATE construction, a role which will be discussed below.

Before further discussion of the relation between valence arguments and slots, we will discuss a somewhat more complex case of valence. The construction we consider is a lexical construction for the word *how* called the MEANS-HOW construction. MEANS-HOW is one of a number of *how* constructions — the HOW-SCALE construction discussed above is another of them, for example, and others include the MANNER-HOW and INSTRUMENT-HOW constructions[11], or the construction in the following quotation from Milne's Winnie-the-Pooh:

> *"And how are you?" said Winnie-the-Pooh.*
> *Eeyore shook his head from side to side.*
> *"Not very how," he said. "I don't seem to have felt at all how for a long time."*

The MEANS-HOW construction is concerned with the *means* of some action, asking for a specification of the means or plan by which some goal is accomplished. It occurs in examples like (3.14a)–(3.14d) (the last three are from *Alice's Adventures in Wonderland*).

(3.14) a. How can I create disk space?

b. The first question of course was, how to get dry again.

c. Let me see–how IS it to be managed?

d. 'Please, then,' said Alice, 'how am I to get in?'

Figure 3.12 shows the representation of the construction.

Like the HOW-SCALE construction, the MEANS-HOW construction includes an **Identify** assertion, with its **Unknown** and **Background** slots. For the MEANS-HOW construction, however, the

[11]Representing the *manner*, *means* and *instrument* senses of *how* as separate constructions is necessary, since besides the semantic distinction, the constructions differ in relative frequency; the MANNER-HOW construction is much less common than the other two senses. Thus note the unacceptability (except in a humorous vein) of (b) vs (c) as a response to (a):

a. How did he clean his room?

b. * — Carefully.

c. — With a vacuum cleaner.

***Means-How* <675**

```
(a Identify $t
      (Unknown $p)
      (Background $x)
 Such-That
  (a Means-For $x
       (Means $p)
       (Goal $g)))

        "how"
```

Figure 3.12: The Means-How Construction

Background element is constrained to be an instance of the **Means-For** concept, while the **Unknown** element is bound to the **Means** slot of this **Means-For** concept. Thus the construction's constitute means that there is a **Means-For** relation which holds between some **Means** and some **Goal**, and the identity of this **Means** is in question.

Having now discussed the valence of a number of different constructions, we turn to a discussion of how valence arguments are *used* and *filled* in processing. Valence arguments specify those aspects of a concept which are *required* to be instantiated in some way. Thus since a lexical construction like CREATE specifies two valence arguments, **Creator** and **Created**, both of these arguments must be instantiated in order for the construction to be felicitous.

The element which instantiates a valence argument can appear in a number of places, locally or distantly. In the case of CREATE, for example, the local SUBJECT-PREDICATE construction may instantiate the **Creator** as the subject of the verb, while the VERB-PHRASE construction may instantiate the **Created** as a direct complement of the verb. Distant instantiation occurs with *focusing* constructions, the gapping constructions referred to as *right-node raising* in the transformational paradigm, or any of the *wh-* constructions, For example the WH-NON-SUBJECT-QUESTION construction may instantiate the **Created** role of the CREATE construction as a *wh*-element, as in (3.15):

(3.15) What did Rodin create?

In (3.15), the **Created** valence slot of the CREATE construction is filled by the semantics of the lexical construction *what*.

Some constructions allow their valence arguments to be filled even more distantly, such as by elements outside the clause or the sentence. Among these are the cases of *null complement anaphora* discussed by Grimshaw (1979) and Fillmore (1986). Since the model of interpretation presented in this dissertation only deals with single sentences, the problems of representing *which* lexical constructions allow null-complement anaphora, and of correctly filling the appropriate valence slot, will not be discussed here.

Since all valence slots correspond to *required* arguments, optional arguments cannot be represented as valence slots. Instead, optional arguments are represented in one of two ways, depending on the argument. Some arguments, such as the optional *that* complementizer in the SUBORDINATE-PROPOSITION construction, are represented by creating two copies of the construction, one with the argument, and one without. This option is chosen in cases where the meaning of the construction seems to differ depending on whether or not the element is present.

Other types of optional arguments, such as the *time* and *location* adjuncts which are often but optionally attached to activity verbs, are represented by allowing the *concepts* for these verbs to have unfilled variables. Placing unfilled variables in the *concept* is different than placing them in the *construction* since, as discussed above, a concept may contain a number of slots which are not mentioned in the construction. Thus the fact that the CREATE construction may optionally appear with a *time* adjunct as in (3.16) is predicted by the **Time** slot in the **Creation-Action** concept.

(3.16) When did she create that sculpture?

This idea of allowing optional time and location adjuncts to appear when they are specified semantically was proposed by Gawron (1983).

This concludes the discussion of CIG. All the constructions which appear in the rest of the dissertation use the representational mechanisms described in this chapter; many of the individual constructions from this chapter will reappear later. The rest of the dissertation will focus on the architecture and sub-theories of the interpreter Sal.

Chapter 4

The Architecture of the Interpreter

Chapter 1 introduced the fundamental problem which we will explore in further detail in this chapter: modeling the process of human sentence interpretation. This chapter describes the architecture of Sal itself and summarizes the *access*, *integration*, and *selection* theories which are examined in detail in Chapters 5, 6, and 7.

4.1 Architectural Principles

This section gives an overview of the interpreter by introducing four architectural principles, and sketching how these principles relate to the architectures of other models. The principles express four properties of the model; it is *on-line*, *parallel*, *interactionist*, and *uniform*. The first architectural principle, the **On-Line Principle**, follows directly from the criterion of cognitive adequacy:

> **On-Line Principle:** *Maintain a continually-updated partial interpretation of the sentence at all times in the processing.*

There is a great amount of psycholinguistic evidence for the on-line nature of interpretation building, including evidence from comprehension (Marslen-Wilson 1975; Potter & Faulconer 1979), lexical disambiguation (Swinney 1979; Tanenhaus *et al.* 1979; Tyler & Marslen-Wilson 1982; Marslen-Wilson *et al.* 1988), pronominal anaphora resolution (Garrod & Sanford 1991; Swinney & Osterhout 1990), verbal control (Boland *et al.* 1990; Tanenhaus *et al.* 1989), and gap filling (Crain & Fodor 1985; Stowe 1986; Carlson & Tanenhaus 1987; Garnsey *et al.* 1989; Kurtzman *et al.* 1991).

The on-line principle has two implications for the interpreter. The first is that it must produce an interpretation incrementally, that is in a strictly left-to-right manner while the sentence is being processed. This rules out the traditional depth-first or backtracking control structure for parsers, because these parsers may make an indefinite number of left-to-right scans over the input. Thus, for example, depth-first ATN's do not conform to the principle. Systems which employ very local backup (such as single word backup) can still be on-line, and hence are not ruled out by the on-line principle.

The second implication of the on-line principle is that the interpreter cannot maintain all possible interpretations of a sentence during the processing. It is required, fairly frequently, to choose a single interpretation with which to continue processing, in accordance with the psycholinguistic evidence present above. This rules out the use of parallel parsers which maintain every possible syntactic or semantic structure in parallel, such as the active chart parser of Kaplan (1973), the breadth-first ATN parser (Woods 1970), or the expanded LR-style parser of Tomita (1987). Indeed Church & Patil (1982) have shown that attempting to maintain every possible syntactic structure for sentences with preposition-phrase ambiguities is extremely difficult.[1]

Unfortunately it is not possible to follow the on-line principle by simply choosing an interpretation immediately whenever an ambiguity arises. This is due to the fundamental conflict in human language understanding between the need to produce an interpretation as soon as possible, and the need to produce the correct interpretation. Because evidence for the correct interpretation may be delayed, any on-line interpreter must choose a method for integrating this late evidence.

Our model uses limited local parallelism to represent these local ambiguities while waiting for further evidence. At any point, multiple possible candidate interpretations are entertained, but only for a short time, and the interpreter is forced to choose among them quickly. We can summarize this as the *Parallel Principle* below:

> **Parallel Principle:** *Keep multiple partial interpretations for a limited time during processing of a sentence.*

There is a great deal of evidence for temporary local parallelism in *lexical* processing (such as Swinney (1979), Tanenhaus *et al.* (1979), and Tyler & Marslen-Wilson (1982)). Cacciari & Tabossi (1988) describe results that provide strong evidence for temporary local parallelism in the processing of *idioms*. Finally, Kurtzman (1985), Gorrell (1987) and 1989, and MacDonald *et al.*

[1] Note that the *On-Line Principle* thus rules out backtracking and long-term parallelism, as does Marcus (1980), but for quite different reasons. Marcus's *Determinism Hypothesis* rule out backtracking and parallelism because they are ways of simulating a non-deterministic machine. The *On-Line Principle* rules out backtracking and long-term parallelism because they make it impossible to produce a single interpretation of a sentence in a on-line fashion.

(in press) present evidence for parallelism in *syntactic* processing. [2] A number of recent models of interpretation use the limited parallel framework, including Gibson (1991), Gorrell (1987), and Kurtzman (1985). §4.8.2 contains a further discussion of these other models.

Other earlier approaches to modeling local syntactic ambiguity have generally been serial rather than parallel models, and fall into two classes. The first class might be called *delayed-choice serialism*, and has been referred to as the "wait-and-see" approach. It was first proposed by Marcus (1980) and is used by other Parsifal-style parsers (Milne (1982), Charniak (1983)) as well as the shift-reduce parser of Shieber (1983), and the Description Theory model of Marcus *et al.* (1983). In these approaches, the model waits to build structure until it can be certain it is building the correct interpretation, although the delay is strictly limited. Pritchett (1988) and Gibson (1991) note a number of problems with these models, such as the incorrect prediction that certain sentences will be unproblematic when they do in fact cause the garden path effect. In general, these problems are caused by the fact that delayed-choice serial models have difficulty correctly specifying exactly how long to delay.

The second class of models implement *immediate-choice serialism* by using global heuristics (such as Minimal Attachment) to resolve local ambiguity immediately. Because such global heuristics are syntactic, immediate-choice serial models are almost invariably *parsing* models rather than models of interpretation. Example of these include Kimball (1973), Frazier & Fodor (1978), Wanner (1980), and Pritchett (1988). These models suffer from a number of problems. First, because the models are limited to syntactic structure, they fail to meet the criterion of functional adequacy introduced in Chapter 1. Next, they are incompatible with psycholinguistic data which supports parallelism in syntactic processing, such as Kurtzman (1985), Gorrell (1987) and 1989, and MacDonald *et al.* (in press). These models are also uneconomical in assuming that lexical processing is done in parallel but syntactic processing is done serially, thus requiring that separate access, integration, and selection mechanisms be used for lexical and non-lexical structures. Finally, the exact specification of these global syntactic heuristics is quite difficult.

The third principle, the **Interaction Principle**, calls for a interactionist, knowledge-based approach to sentence processing.

> **Interaction Principle:** *Make use of syntactic, semantic, and higher-level expectations to help access linguistic information, integrate it into the interpretation, and choose among candidate interpretations.*

Interactionist architectures are quite widespread in natural language processing models, such as Wilks (1975a), Riesbeck & Schank (1978), Cullingford (1981), Phillips & Hendler (1982), and Adriaens & Small (1988). But most theories that explicitly attempt to model human sentence processing have avoided expectation-driven processing. Sentence processing models all assume that contextual information is used at some point in processing; the disagreement is over whether this high-level knowledge can be used early in the access and integration of linguistic knowledge.

Most models have particularly avoided the use of expectations to suggest lexical items or syntactic constructions in a top-down way. This use of expectations, called variously "strong

[2] By *parallelism* we mean what Ward (1991) called 'competitive parallelism' — simultaneous consideration of several alternative interpretations. The interpreter does not employ what Ward called 'part-wise parallelism' — working on several words of the input in parallel. Words are still input in a serial fashion.

interactionism" or "contextual preselection", is a characteristic of our model. An interactionist model (such as McClelland 1987) allows information from any level of linguistic processing to affect any other; in particular, semantic knowledge may directly affect the access of lexical or syntactic constructions. Sal allows the use of frames (like Hirst 1986), thematic roles (like Carlson & Tanenhaus 1987 and Stowe 1989), and other high-level semantic information (like Riesbeck & Schank 1978) to build interpretations.

Other models fall into two classes. Models which fall in the first, "modularist" or "non-interactionist" class (Frazier 1987b, Clifton & Ferreira 1987, Clifton & Ferreira 1989) consist of highly autonomous processing modules which are informationally encapsulated. Syntactic processing, for example, is done by a syntactic module which is insensitive to semantic or other high-level effects.

The second class of models, the "weak interactionist" models, have been described as the "syntax proposes and semantics disposes" models. Here higher levels may help choose among the output of lower levels, but may not act as to pass information to these levels. Weak interactionist models include Crain & Steedman (1985), Marslen-Wilson (1987), Tyler (1989), Steedman (1989), Tanenhaus & Carlson (1989), Cottrell (1989), and Altmann (1988).

In general, psycholinguistic evidence has not been conclusive in deciding among the *strong-*, *weak-*, and *non-interactionist* positions. While such studies as Swinney (1979) and Tanenhaus *et al.* (1979) initially argued that lexical access was independent of contextual influences, Simpson (1984) and McClelland (1987) showed that even these studies displayed slight effects of context. More recent studies such as Tabossi (1988) and Simpson & Kellas (1989) have found interactionist effects by using particularly strong contexts. While modularist models such as Frazier (1987b) have used the existence of garden-path effects to argue for modularism, models such as those of Crain & Steedman (1985) and Altmann & Steedman (1988) have shown that garden-path effects can be accounted for in an interactionist architecture. The issue of interactionism is discussed in more detail in §5.5.

The final principle, the **Uniformity Principle**, makes more-specific claims about the algorithm used to produce the interpretation.

> **Uniformity Principle:** *A single interpretation mechanism accounts for the access, integration, and selection of structures at all levels of sentence processing*

The Uniformity Principle proposes a single, integrated mechanism to replace the traditional informationally encapsulated lexical analyzer, syntactic tree-builder, morphological analyzer, and interpretation mechanisms. Recall that the **Grammatical Construction Principle** of Chapter 3 proposed that a single representational device account for all linguistic knowledge — lexical items, idioms, constructions. The Uniformity Principle extends this representational uniformity to the processing domain. Thus our model does not require a separate lexicon, grammar rule-base, idiom dictionary, and semantic interpretation rule-base, nor the various processing mechanisms each would need.

A corollary of the Uniformity Principle is that syntactic and semantic processing are not distinct; the model does not distinguish the *parser* from the *semantic interpreter*, or indeed from the *lexical analyzer*. The functions of access (proposing constructions to use in an interpretation), integration (combining constructions to build an interpretation), and selection (choosing among interpretations) apply uniformly across the lexical, syntactic, and semantic domains. For example, the access function accounts for the access of lexical items as well as syntactic rules (this is natural because both are represented as *grammatical constructions*). The integration function builds structures by combining component structures at each level (in building words, syntactic phrases, or semantic interpretations). The selection function resolves both lexical and higher-level ambiguities.

It is important to note that integrating syntactic and semantic processing does not mean ignoring one paradigm or the other. As Hirst (1986:2) has noted, “those who argue for the integration of syntactic and semantic processing are usually disparaging the role of syntax”. The criterion of representational adequacy introduced in Chapter 1 required that *both* syntactic and semantic knowledge be adequately represented. Indeed, the fact that syntactic and semantic knowledge are *uniformly* represented in grammatical constructions makes it quite natural that the interpreter give equal consideration in processing to each kind of knowledge.

Psycholinguistic evidence for the uniformity principles arise from results which show that the functions of access, integration, and selection apply uniformly to lexical, idiomatic, and syntactic structures. For example, the studies cited in support of the Parallel Principle above showed that lexical, idiomatic, and syntactic structures are all accessed and maintained in parallel. Other evidence shows that the access of structures at all levels is sensitive to context and multiple knowledge sources (Salasoo & Pisoni 1985, Cacciari & Tabossi 1988, and Marslen-Wilson *et al.* 1988). Chapter 7 shows that a uniform selection theory can account for lexical, idiomatic, and syntactic preferences in disambiguation.

The Uniformity Principle distinguishes Sal from the majority of sentence-processing models, which draw especially sharp distinctions between syntactic and semantic processing. Such models include those associated with theories of grammar, such as Ford *et al.* (1982) (LFG), Proudian & Pollard (1985) (HPSG), the Government and Binding parsers such as Pritchett (1988), Abney (1989), Johnson (1991), or Fong (1991), or models such as Frazier & Fodor (1978), as well as a number of AI models such as Winograd (1972), Mellish (1983), and Hirst (1986).

As we noted above following Hirst, models which attempt some level of uniformity of syntactic and semantic processing have generally given short shrift to syntactic knowledge. These include models such as Riesbeck & Schank (1978), Wilensky & Arens (1980), and Cater (1983).

4.2 Introducing the Algorithm

Like Caesar’s Gaul, Sal’s architecture consists of three components: the *working store*, the *long-term store*, and the *interpretation function*.

- The *working store* contains constructions as they are accessed, and partial interpretations as they are being built up. It consists of two data structures: the **access buffer** and the **interpretation store**. When a construction is first accessed, it is copied into the **access buffer**, and is then integrated into the **interpretation store**, which may contain a number of partial interpretations. The working store is constrained in the *number* of interpretations that it can hold, and by the *time* that it can hold them. In this way it models the similar limitations of human short-term memory which have been shown to place specific constraints on interpretations (Gibson (1991), MacDonald *et al.* (in press), etc.). Limitations on short-term memory affect both the access buffer and the interpretation store.

- The *long-term store* contains the linguistic knowledge of the interpreter (i.e., the grammar). This knowledge consists of the collection of grammatical constructions discussed in Chapter 3. The *long-term store* also includes the representation of general, non-linguistic knowledge.

- The *interpretation function* is the processing component of the interpreter. As Chapter 1 discussed, any theory of interpretation processing must include sub-theories of **access**, **integration**, and **selection**. Thus the interpretation algorithm will be discussed by describing a *control structure* and the three functions of access, integration, and selection. When the interpreter is given a sentence as input, it first relies on the **access** function to amass evidence for constructions in the grammar, and to copy suggested structures into the access buffer. The **integration** function then integrates these structures together to produce *candidate interpretations* in the **interpretation store**, and the **selection** function chooses a single interpretation among the candidate interpretations.

Figure 4.1 presents a schematic diagram of the architecture, showing each of the functions and each of the data structures.

The control algorithm for the interpreter simply calls each of the three functions to do the appropriate manipulation of interpretations. The algorithm can be sketched abstractly as follows; the details of access, integration, and selection will be discussed afterwards.

1. Examine the input. As evidence accumulates for the applicability of constructions in the grammar, increase their activation values.

2. When a construction's activation passes the *access point*, copy it into the access buffer, or if the construction was suggested by evidence already in the access buffer, *integrate* it directly with the access buffer.

3. Integrate the access buffer with the interpretation store as follows (successful integration may increase the size of the buffers):

   For each interpretation i in the interpretation store

       Make a copy c of the interpretation i

       For each construction a in the access buffer

           Integrate the current point (the cursor) of c with a.

   Clean up by removing any structures which failed to integrate.

4. Clear out the access buffer after integration.

5. Update the selection rankings of each interpretation in the interpretation store

6. If any interpretations in the interpretation store are worse than the best interpretation, by at least the selection threshold σ, prune them from the interpretation store. If only one interpretation remains in the selection store, it is *selected*.

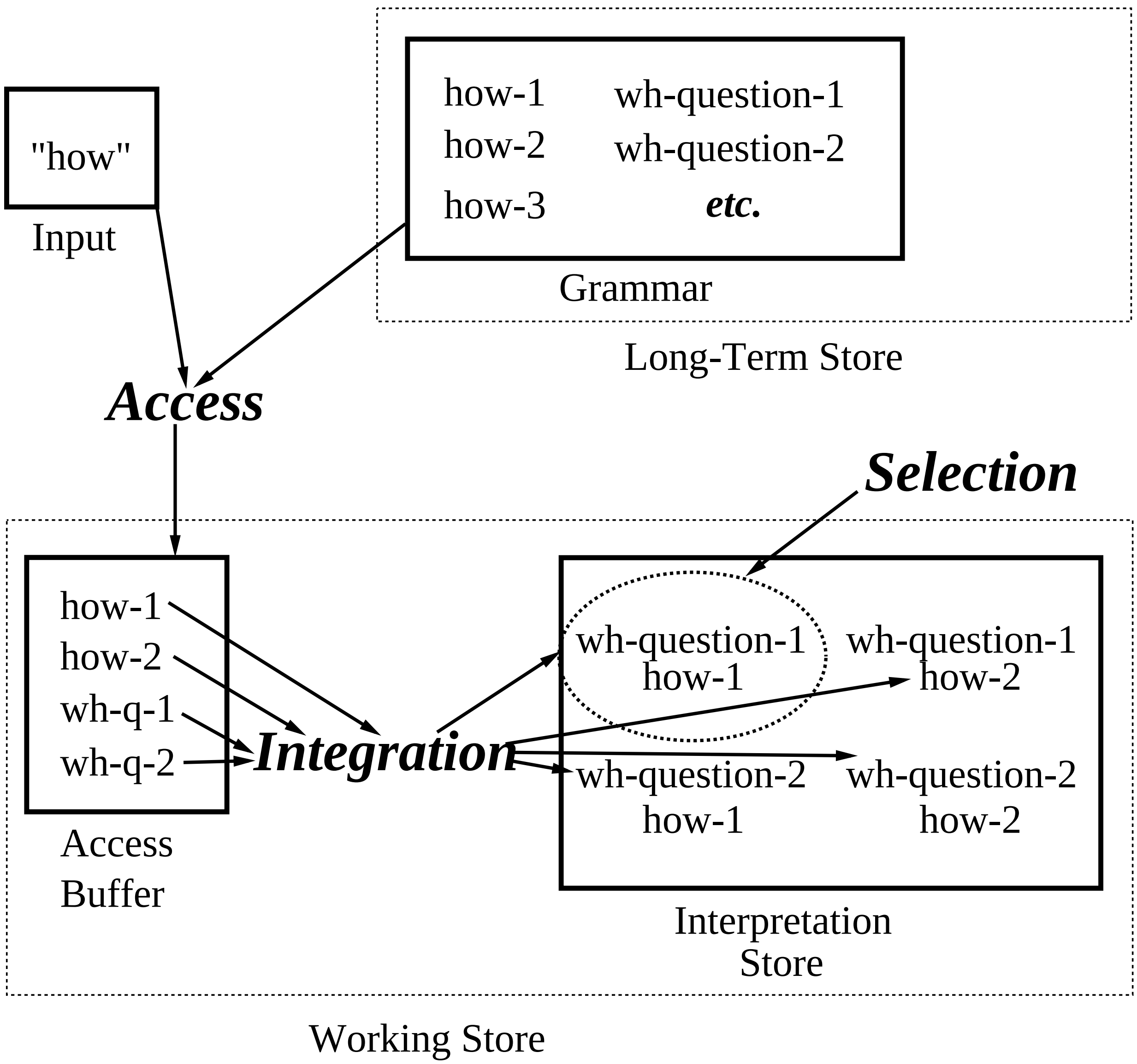


Figure 4.1: The Architecture of the Interpreter

7. Go to 1.

The next five sections will further describe each of these functions and data structures. Even more detailed discussion of each is contained in Chapters 5, 6, and 7. §4.7 shows how the cognitive constraints on the interpreter allow it to avoid the well-known complexity and efficiency problems. Finally, §4.9 provides a short trace of the processing of a sample sentence.

4.3 The Access Theory

The first function, the **access** function, must decide when to copy a construction into the access buffer. The access function and the algorithm which implements it can be specified as follows:

> **Access Function:** Access a construction whenever the evidence for its applicability passes the access threshold α.

Access Algorithm:

1. Each construction in the grammar has an *activation value*, which is initialized to zero.

2. As the interpreter encounters evidence for a given construction, the activation value of the construction is increased by the number of "access points" corresponding to the new evidence.

3. When the activation value for a construction passes the access threshold α, a copy of the construction is inserted in the access buffer. This point in time is called the "access point",

4. After each access round, the activation value of each construction in the grammar is reset to zero.

The access algorithm shares a number of properties with the interpreter as a whole. First, the access algorithm is *uniform*. Since all linguistic information (lexical items, idioms, syntactic rules, semantic rules) is represented uniformly as *grammatical constructions*, a single access algorithm can access all this information uniformly. Each type of constructions is annotated with relative frequencies, and higher-frequency constructions are more likely to be suggested. The access algorithm is *parallel*, in that it suggests and activates multiple grammatical constructions at a time. Each construction whose activation value passes the access threshold α is inserted in the access buffer. Thus if multiple constructions pass the **access point** simultaneously, the access buffer will contain a number of constructions at a time, The access algorithm is *on-line* in accumulating evidence for the access of each constructions continuously and incrementally. As the interpreter amasses evidence for a construction, it adds the evidence values (expressed in "access points" which are proportional to the frequency of the construction — see §5.2) to the current state of each construction.

Finally, access is *interactionist* in using any kind of linguistic information, including top-down or contextual information, to provide evidence for accessing constructions. Knowledge sources that the access function allows include:

- *Bottom-up syntactic evidence:* For example, the fact that a construction's first constituent matches the contents of the access buffer is evidence for that construction.
- *Bottom-up semantic evidence:* evidence for a construction whose left-most constituent matches the semantic structures of some structure in the access buffer.
- *Top-down syntactic evidence:* when a construction's *constitute* matches the current position of some construction in the interpretation store.
- *Top-down semantic evidence:* when a construction's *constitute* matches the semantics of the current position of some construction in the interpretation store, or matches the semantic expectations of a previously encountered lexical item.

Chapter 5 will give examples of each of these kinds of evidence. Figure 4.2 shows the state of the system, including the grammar, the input, and the access buffer, after seeing the word *how*. The activation value which is associated with each construction in the grammar is represented in Figure 4.2 by an activation meter.

Note in Figure 4.2 that the word *how* has provided evidence for two constructions. The first construction (the MEANS-HOW construction) is a lexical one, and is the sense of "how" which is concerned with specifying the means or plan by which some goal is accomplished ("how can I get home?"). The second (called **How-Scale**) expresses a question about some scalar properties ("How red is that dress?"). Both constructions are discussed in Chapter 3. The representation of the semantics of the two constructions is somewhat difficult to read; we are perhaps more familiar with descriptions of parsers which manipulate the traditional **N**'s and **V**'s. The complex diagrams in the figures in this chapter are an unfortunate side-effect of the fact that neither CIG nor the interpreter assume an autonomous syntax, and thus syntactic and semantic constraints are represented and interact at the same level. In any case, see §3.8 for a description of the semantic language used in the examples in this dissertation.

When a construction is accessed, it is either *copied into* the access buffer or *integrated with* the access buffer. Which of these is done depends on how the construction was accessed. If a construction is accessed by bottom-up evidence from a construction that is *already* in the access buffer, it is integrated with that construction. For example when a lexical construction c is accessed, it would provide evidence for a construction f whose first constituent can be c. However, when f is accessed from bottom-up evidence, it cannot simply be inserted into the access buffer, because the access buffer already contains c. Instead f is *integrated* directly into the access buffer. Thus in the case of bottom-up evidence f and c are integrated together *before* they are integrated into the interpretation store.

4.4 The Interpretation Store

We have discussed the access function, and its data structure the access buffer. Before discussing the integration function, we turn to the data structure in which candidate interpretations are kept, the **interpretation store**.

After constructions appear in the access buffer, they are integrated with the interpretations in the interpretation store. The interpretation store contains a disjunction of interpretations which

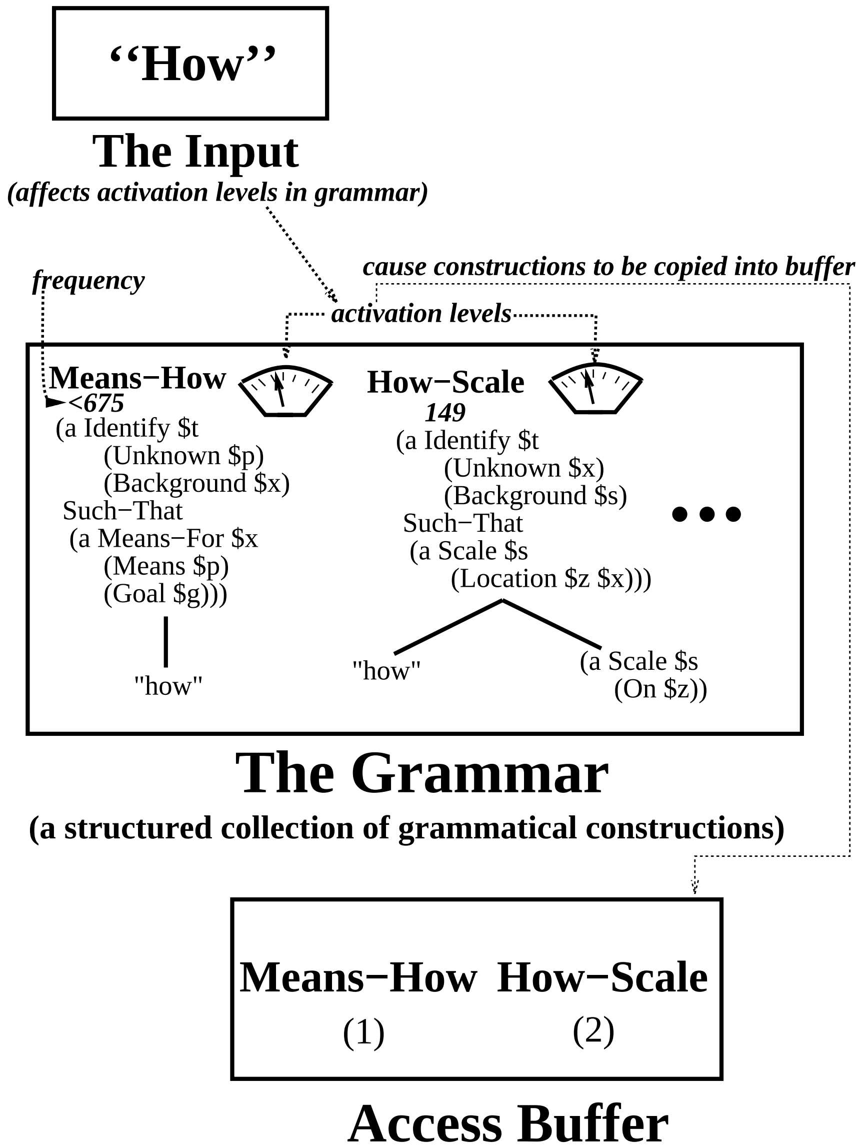


Figure 4.2: The access buffer after seeing "how"

account for all the input which the interpreter has seen at the current point in the interpretation process. Figure 4.3 shows a schematic display of the access buffer with four constructions and the interpretation store with two possible interpretations.

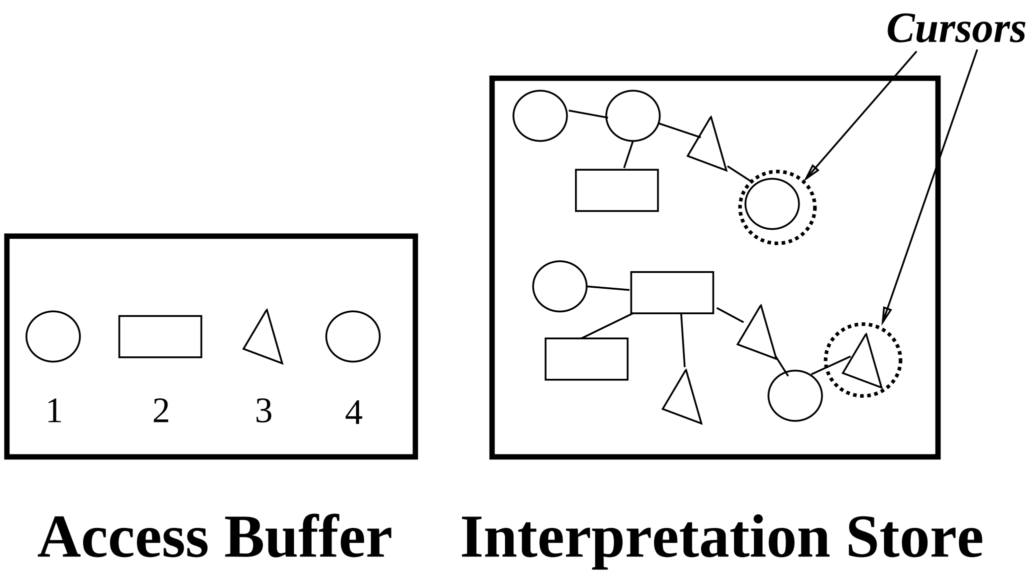


Figure 4.3: The access buffer and the interpretation store

For explanatory purposes, Figure 4.3 uses geometric shapes to stand for different constructions which may occur in the access buffer or in the interpretation. Each interpretation in the interpretation store consists of a number of constructions (combined by the integration mechanism discussed below), and thus Figure 4.3 depicts an interpretation as a network of these geometric shapes.

Note that one of the constructions in each partial interpretation is specially marked by encircling it with a dotted line. This mark indicates the *cursor*, which is the current position of the interpreter in each interpretation. For each interpretation, its cursor indicates the location in the interpretation at which the integration control process will attempt to integrate the contents of the access buffer. As the interpretation process proceeds, the integration control will fill in the constituents of each interpretation one by one. As this happens, the cursor for each interpretation moves forward each time to the next constituent. In the remainder of this chapter and in the following chapters, whenever processing examples are shown in figures, the cursor will be marked by a dotted line circling the constituent.

In Figure 4.3, the particular geometric shape which is marked by the cursor indicates the constraints on the construction that is to fill this slot. These constraints can be syntactic, semantic, or pragmatic. The next section and Chapter 6 show how these constraints are exploited by the integration algorithm.

The use of a device for marking the current point in the interpretation is universal to parsers and interpreters. Traditional parsers do this in various ways — LR parsers with the *dot* in items, chart parsers with open chart edges, Marcus parsers with the constituent in the first buffer, ATN's by the top of the execution stack, conceptual dependency analyzers by maintaining a marked 'now-point'. The term *cursor* was first used by Ward (1991) in his model of natural language *generation* to describe the current point in a grammatical construction as it is being generated.

4.5 The Integration Theory

Any interpreter must have a way to build up interpretations from (among other things) their component constructions. We call the part of the theory which instantiates this process the *integration theory*. Integration is the process by which the meaning of a construction and its various constituents are incrementally combined into an interpretation for the construction. We may conveniently divide the integration theory into an *integration control structure* and an *integration operation*. The control structure specifies how the interpreter attempts to integrate each of the constructions in the access buffer into each of the interpretations in the interpretation store. Recall that Sal's control structure as described in §4.2 called the integration operation as follows:

For each interpretation i in the interpretation store
    Make a copy c of the interpretation i
    For each construction a in the access buffer
        Attempt to integrate the cursor of c with a.
Cleanup by removing any structures which failed to integrate.

The integration operation itself is thus called on each interpretation-construction pair, and attempts to integrate each construction with the cursor of each interpretation.

4.5.1 The Integration Operation

In introducing the integration operation itself, it is important to note its limitations. In particular, the integration operation is not intended to model the entire process of interpretation-building. We may divide this large problem into two components — *grammaticalized combination* and *inferential combination*. The integration operation only solves the first of these problems — combining meanings when the means or nature of the combination is specified in some grammatical construction.

Integration was designed as an extension to the *unification* operation (Kay 1979). While unification has been used very successfully in building *syntactic* structure, extending the operation to building more complex *semantic* structures requires three major augmentations:

- The integration operation includes knowledge about the representation language which is used to describe constructions (see §3.8). This allows the interpreter to use the same semantic language to specify constructions as it uses to build final interpretations, without requiring translation in and out of feature structures. The integration operation can also use information about the representation language to decide if structures should integrate; thus it will integrate two constructions if one *subsumes* the other.

- The integration operation distinguishes *constraints* on constituents or on valence arguments from *fillers* of constituents or valence arguments.
- The integration operation is augmented by a *slash* operator, which allows it to join semantic structures by embedding one inside another. This is accomplished by finding a semantic gap inside one structure (the *matrix*), and binding this gap to the other structure (the *filler*). This operation is similar to the *functional-application* operation and the *lambda-calculus* used by other models of semantic interpretation.

This integration operation is used in two ways in building interpretations, *constituent integration* and *constitute integration*. *Constituent integration* is the process by which a construction's constituent slots are filled by other constructions. In order to fill a constituent slot, a candidate filler must meet the constraints imposed on that slot by the construction. *Constitute integration* is the process by which the semantics of each of these constituents is combined to build an interpretation. Constitute integration may be as simple as linking semantic structures by co-indexing a variable, or may involve more complex combinations of structures.

Constituent integration is very much like a more fine-grained version of the *handle-pruning* mechanisms used by bottom-up parsers (Aho *et al.* 1986). Informally, a *handle* is a substring of the input that matches the right-hand side of some rule. Handle-pruning thus consists of replacing a handle in a string with the left-hand side of the relevant rule. In constituent integration, instead of matching the entire right-hand side of a rule with the input, we match a *single constituent* with the input. Integration thus proceeds on a *constituent-by-constituent* basis, instead of the *rule-to-rule* basis which is used in many models of sentence-interpretation as well as in many parsers used for programming languages. This is discussed further in §6.2.3. The constituent integration algorithm is specified as follows:

> **Constituent Integration Algorithm**: Given a construction c which places a set of constraints s on its cursor constituent, and given a proposed constituent g, integrate each assertion in g with each assertion in s, subject to the constraint that s must subsume g.

Figure 4.4 illustrates the constituent integration algorithm. The interpretation store contains a construction whose cursor is specified to be a VERB. The access buffer contains a construction which is in fact a VERB. Thus the constituent integration algorithm will integrate the construction in the access buffer with the construction in the interpretation store. See §6.4.2 for more detailed examples using real constructions.

Unlike constituent integration, the *constitute* integration algorithm is not called by the integration control algorithm, but rather is called whenever a constituent of a construction has just been integrated, and the construction itself specifies that the bindings of certain variables should be integrated. If a construction specifies that one structure should be bound to a *hole* inside another, as does the VERB-PHRASE construction, or the DETERMINATION construction, constitute integration calls the *valence integration* algorithm. A sketch of the valence integration algorithm follows; details are discussed in Chapter 6.

> **Valence Integration Algorithm:** Given a matrix variable m and a filler variable f, examine each hole h_i in m, and when the constraints on a given hole h_n meet the

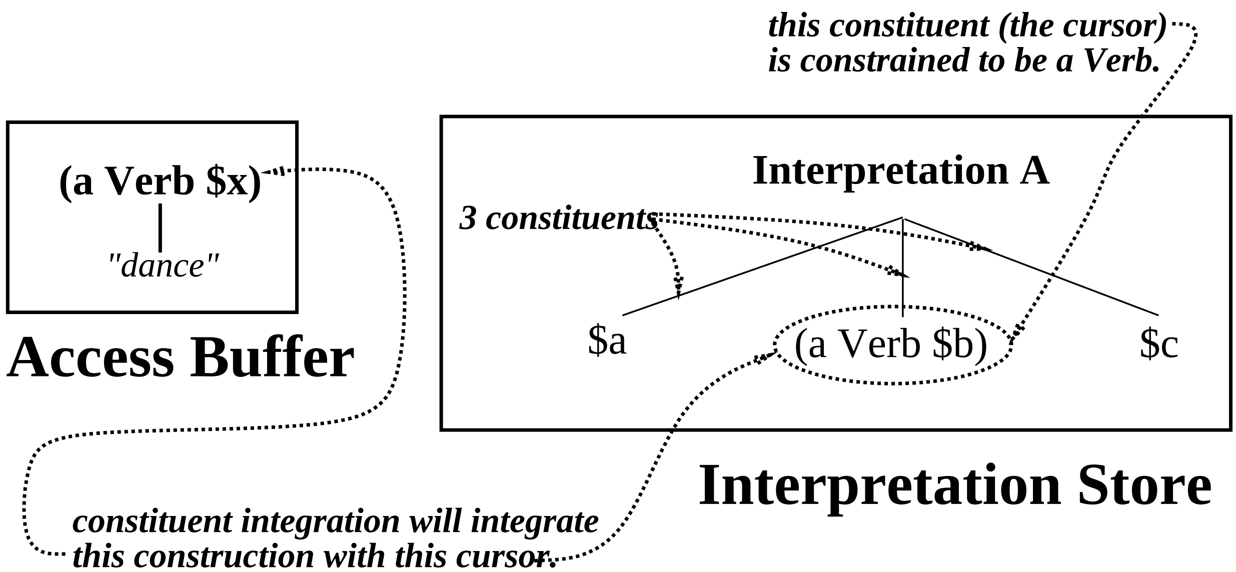


Figure 4.4: Constituent Integration

constraints on the filler f, integrate h_n with f. If there is no such hole h_n, but some part of the matrix m is still incomplete, wait and try again.

4.5.2 An Example of Integration

This section illustrates the concept of integration by showing an example of a single integration step. The example takes place in processing the sentence fragment:

(4.1) Peter will can. . .

This fragment is useful as an example, even if a bit contrived, because of the multi-categorial ambiguities of *can*. *Can* can be a noun, an auxiliary, or a verb (in fact two verbs, one meaning “to put in a can”, the other “to dismiss an employee”). In the example above, the sentential context is only compatible with the verbal reading of *can*. Thus (4.1) the sentence might be completed as in (4.2a) or (4.2b):

(4.2) a. Peter will can all this salmon by 5:00.

b. Peter will can that employee who was accused of insider trading.

Figure 4.5 displays the access buffer after the access of the various lexical *can* constructions, and the interpretation store with the relevant part of the interpretation.

In the example shown in Figure 4.5, each of the four structures in the access buffer will be integrated with the structure in the interpretation store. If the partial interpretations did not place any constraints on the constructions which may integrate into them, the number of possible

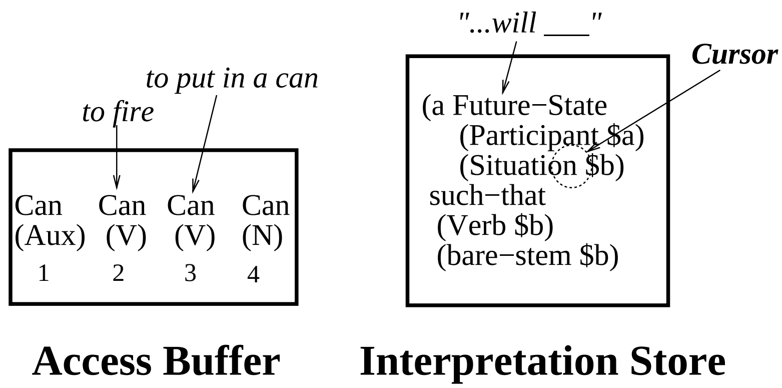


Figure 4.5: Interpreter State after seeing *can*

interpretations would grow from one to four. A copy of the interpretation is made for and then integrated with each sense of *can*. If the interpretation store had begun with two interpretations, it would grow to 8. In general, each time the access buffer is filled, integrating the disjunction of accessed constructions with the disjunction of partial interpretations would cause the size of the interpretation store to increase to the product of the sizes of the store and the buffer.

However, because each interpretation imposes constraints on which constructions may integrate with it, some of the possible combinations may be ruled out in integration. For example, the result of integrating the access buffer of Figure 4.5 into the interpretation store would result in a new interpreter state shown in Figure 4.6.

Note that in Figure 4.6 the interpretation store only contains two interpretations, rather than four. This is because the other two possible interpretations were ruled out by constraints on the integration. These constraints were placed on the cursor of the single interpretation in the interpretation store in Figure 4.5. Note that the verb ”will” required that its complement be the stem-infinitive form of a verb. This constraint ruled out the nominal sense (sense 1) as well as the auxiliary sense (sense 4) of *can*. (The latter because auxiliaries have no non-finite forms). This left two constructions in the access buffer — the two verbal senses of *can*[3]. Each of these senses can be integrated with the interpretation store, resulting in two partial interpretations.

We have seen that the integration mechanism applies both syntactic and semantic constraints in an on-line fashion. Chapter 6 will present further examples of integration, as well as showing examples of the gap-finding algorithm and discussing psycholinguistic evidence for integration.

4.6 The Selection Theory

Any model of interpretation which allows parallel structures must include a theory for choosing among these structures. We call the function which instantiates this theory the **selection** function.

[3]Both verbal senses of *can* are actually ambiguous between the stem-infinitive and the non-third-person-singular present tense finite form. We have ignored this second sense of each verb, since it is also ruled out by the stem-infinite constraint placed by the verb *will*.

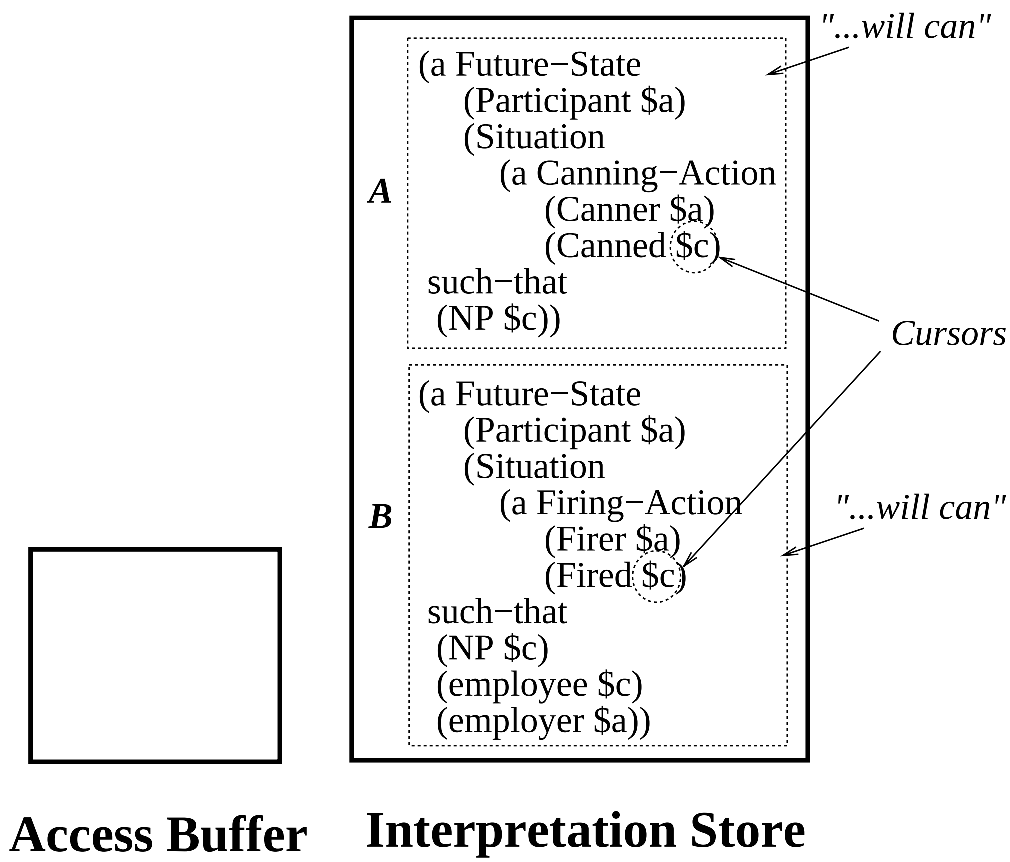


Figure 4.6: After Integration

The selection function chooses an interpretation from the disjunction of candidate interpretations in the interpretation store. Any selection theory must answer two fundamental questions:

> *How* to choose among the interpretations?
> *When* to select an interpretation?

Sal solves the first problem by ranking the interpretations according to a metric, and selecting the most-favored interpretation by this metric. The metric the interpreter uses is *coherence with expectations*, and the theory assigns preferences to interpretations by the **Selection Choice Principle**:

> **Selection Choice Principle:** Prefer the interpretation whose *most recently integrated element* was the most *coherent* with the interpretation and its lexical, syntactic, semantic, and probabilistic expectations.

The Selection Choice Principle refers to a number of kinds of expectations. The term "expectation" has been used most frequently to mean the sort of slot-filling processing that is associated with the scripts of Schank & Abelson the frames of Minsky, or the schemata of Bartlett. The term is used for similar purposes in the selection theory. Selection theory expectations include *constituent expectations*, which are expectations which a grammatical construction has for particular constituents, *valence expectations*, which are expectations that particular lexical items have for their arguments, as well as *frequency expectations*, based on the idea mentioned in Chapter 5 that more frequent constructions are more expected than less frequent constructions, all things begin equal. As Chapter 3 showed, each construction is annotated with a relative frequency, drawn from its occurrence frequency in the Brown Corpus. Thus an expectation is defined as any structural constraint placed by previously-encountered linguistic structures which can help narrow down the search space for predicting or disambiguating the structures which follow.

The coherence of a recently integrated element with an interpretation is defined according to the following ranking:

> **The Coherence Ranking** (in order of preference, with coherence points in parenthesis):

I*(3)* Integrations which fill a *very strong* expectation such as one for an exact construction, or for a construction which is extremely *frequent*.

II*(1)* Integrations which fill a *strong* expectation such as a *valence expectation* or a *constituent expectation*.

III*(1)* Integrations which fill a *weak expectation*, such as for an optional adjunct or include feature matching rather than feature imposing.

IV*(1)* Integrations which fill no expectations, but which are nonetheless successfully integrated into the interpretation.

V*()* Integrations which are *local*, i.e., which integrate the elements which are the closest together.[4]

VI*(0)* Integrations which fill no expectations, and are not integrated into the interpretation.

Thus when choosing between two interpretations, the selection function will look at the most recently integrated element, and select the interpretation whose ranking is highest on the *Coherence Ranking*. (The numbers after the ranking will be used by the Selection Timing Principle below).

Of course, the selection choice principle will not be sufficient to solve every case of disambiguation — clearly disambiguation is a process that must refer to every level of linguistic knowledge, including pragmatic and textual knowledge which is not considered in this thesis, as well as non-linguistic world knowledge. But the use of coherence as a preference metric provides a framework with which to express the effect of these kinds of knowledge on the interpretation.

The interpreter solves the second problem (*when* to choose) by assuming that because the interpreter's *working store* is limited like human short-term memory, interpretations are pruned whenever they become significantly less-favored than the most preferred interpretation. §4.7 will show that forcing selection to be *on-line* in this manner also solves some long-standing efficiency problems in parsing. The timing constraint is stated in the Selection Timing Principle:

---

[4]The current implementation of Sal has no point value assigned to locality.

> **Selection Timing Principle:** Prune interpretations whenever the difference between their ranking and the ranking of the most-favored interpretation is greater than the selection threshold σ.

The Selection Timing Principle requires that an interpretation is pruned whenever there exists a much better interpretation. When all of the alternative interpretations have been pruned, the most-favored interpretation will be selected. Thus the interpretation store may temporarily contain a number of interpretations, but these will be resolved to a single interpretation quite soon. The point at which one interpretation is left in the interpretation store is called the *selection point*. Like the *access point* of Chapter 5, the selection point is context dependent. That is, the exact time when selection takes place will depend on the nature of the candidate interpretations and the context. Just as the *access threshold* α was fixed but the *access point* was variable, the *selection threshold* σ is fixed, while the *selection point* will vary with the context and the construction.

Specifying selection timing consists of choosing the selection threshold σ in terms of the *Coherence Ranking* above. We propose that the threshold σ be set at **2** coherence points, where coherence points are the numbers which were assigned to the *Coherence Ranking* above.

Chapter 7 shows that this selection algorithm is sufficient to handle most cases of local ambiguity by discussing a number of well-known cases and showing how the algorithm would choose the correct structure in each case. These include various kinds of lexical ambiguity and structural ambiguity such as preposition or adverbial attachment, or the ambiguity between pronominal and extraposition uses of the pronoun *it*.

4.6.1 Psycholinguistic Evidence

Sal is consistent with a number of psycholinguistic results. This section briefly summarizes a number of these results — further details can be found in Chapters 5–7.

There is psycholinguistic evidence supporting parallelism in the access mechanism for the access of many types of linguistic structures: *lexical* (Swinney (1979), Tanenhaus *et al.* (1979), and Tyler & Marslen-Wilson (1982)), *idiomatic* (Cacciari & Tabossi (1988)), and *syntactic* (Kurtzman (1985), Gorrell (1987) and (1989), and MacDonald *et al.* (in press)).

A number of studies have found evidence for top-down and contextual effects on access. Wright & Garrett (1984) found that very strong syntactic contexts can speed up the access of nouns, verbs, and adjectives. Salasoo & Pisoni (1985) found that top-down effects, both syntactic and semantic, can cause constructions to be accessed. There are a number of results suggesting that contextual evidence can speed up access. These include Cacciari & Tabossi (1988) for idioms, as well as lexical studies such as Prather & Swinney (1988), Tabossi (1988), Oden & Spira (1983), and Simpson & Kellas (1989).

Many studies have demonstrated the need for the use of frequency evidence in access. Studies have shown that high-frequency lexical items have higher initial activation than low-frequency ones (Marslen-Wilson (1990)), are accessed more easily (Tyler 1984 and Zwitserlood 1989), and reach recognition threshold more quickly (Simpson & Burgess 1985 and Salasoo & Pisoni 1985).

There are two classes of evidence for the context-dependent *access point* assumed by our theory. The first class, evidence that access is not immediate, includes Swinney & Cutler (1979) and Cacciari & Tabossi (1988) for idioms, and Tyler (1984) and Salasoo & Pisoni (1985) for lexical items. The second type of evidence indicates that the access point is variable even for a single construction, in different contexts. Cacciari & Tabossi (1988) showed that access of idioms was faster in the presence of context. Salasoo & Pisoni (1985) showed the same for lexical constructions. Marslen-Wilson *et al.* (1988) showed the negative case — that that anomalous contexts can slow down the access point of lexical constructions.

The next class of results deal with the timing of integration. There is a great deal of evidence for the on-line, constituent-by-constituent nature of the integration process. This includes evidence from comprehension (Marslen-Wilson 1975; Potter & Faulconer 1979), lexical disambiguation (Swinney 1979; Tanenhaus *et al.* 1979; Tyler & Marslen-Wilson 1982; Marslen-Wilson *et al.* 1988), pronominal anaphora resolution (Garrod & Sanford 1991; Swinney & Osterhout 1990), verbal control (Boland *et al.* 1990; Tanenhaus *et al.* 1989), and gap filling (Crain & Fodor 1985; Stowe 1986; Carlson & Tanenhaus 1987; Garnsey *et al.* 1989; Kurtzman *et al.* 1991).

A very broad class of results supports the *knowledge-intensive* nature of Sal's integration theory, showing that integration makes use of many kinds of information, included syntactic category and subcategory (Mitchell & Holmes 1985), lexical semantic information (Shapiro *et al.* 1987), and verbal control information (Boland *et al.* 1990; Tanenhaus *et al.* 1989). Sal's valence integration algorithm, in which valence-filling takes place semantically at the valence-bearing predicate, is consistent with the results of Crain & Fodor (1985), Stowe (1986), Swinney & Osterhout (1990), and Garnsey *et al.* (1989) that wh- antecedents are filled directly at the verb, those of Tanenhaus *et al.* (1985), Clifton *et al.* (1984), and Tanenhaus *et al.* (1989) that verbal valence information such as the number of arguments is taken into account in gap-filling, and those of Boland *et al.* (1990), Tanenhaus *et al.* (1989), Boland *et al.* (1989), and Kurtzman *et al.* (1991) that the interpreter uses semantic information about the filler (such as animacy) to decide which argument a gap should fill.

Finally, a number of recent results support the use of syntactic and semantic expectations in selection. Trueswell & Tanenhaus (1991) show that garden path effects could be reduced by manipulating the tense of the clause, indicating that temporal information is used by the selection mechanism. Pearlmutter & MacDonald (1991) and Taraban & McClelland (1988) demonstrate similar effects for thematic roles, showing that selection is sensitive to thematic information.

4.7 The Complexity of Interpretation

Many researchers have noted that without some special attempts at efficiency, the problem of computing syntactic structure for a sentence can be quite complex. Church & Patil (1982) showed, for example, that the number of ambiguous phrase-structure trees for a sentence with multiple preposition-phrases was proportional to the Catalan numbers, while Barton *et al.* (1987) showed that the need for keeping long-distance agreement information and the need to represent lexical ambiguity together make the parsing problem for a grammar that represents such information NP-complete. It might seem that the problems of computing *interpretations* would be even more complex, as the interpreter must produce a semantic structure as well as a syntactic one. In fact, we argue that these complexity problems do not arise, specifically because of the cognitive constraints on the interpreter.

The most popular solution to the problem of maintaining multiple parses of an ambiguous sentence, while still parsing in polynomial time, involves dynamic programming techniques. Essentially, the parser stores the common sub-parts of multiple parses, allowing sub-parses to be only represented once, instead of once per parse tree. This method goes by a number of names, and has been proposed a number of times. It was first proposed as the *well-formed substring table* (WFST) by Kuno (1965), as a data structure which stores the results of all previous computations. It then appeared independently as the *chart parsing* algorithm of Kay (1973), and the *Earley* algorithm of Earley (1970). (Sheil (1976) showed the equivalence of the WFST and the Earley algorithm.) More recently, Tomita (1987) recast the algorithm in a bottom-up form, using as his data structure a generalization of the chart or WFST, the *graph-structured stack*. Norvig (1991) shows that all these algorithms can be captured by wrapping the *memoization* operation around a simple parser.

Unfortunately, these solutions to the ambiguous parse tree problem may not generalize to the problem of interpretation. For example, if two parse trees both include an NP, the dynamic programming algorithm can simply store the NP once, because the internal structure of the NP is irrelevant to the global parse. But if two *interpretations* share the same NP, it may not be possible to store the NP only once, because its internal structure, and particularly its semantic structure, is relevant to the interpretation, and may be needed by the interpreter to produce part of an on-line interpretation. Building the semantics of the NP into the interpretation may involve binding variables differently in the context of different interpretations. Although some semantic structure can most likely be shared, the sharing will not be as efficient as for syntactic structure.

Sal uses another method of avoiding complexity problems. Note that the results of Church & Patil (1982) and Barton *et al.* (1987) rely on the fact that syntactic ambiguities in these parsers are not resolved until *after* the entire sentence has been parsed. It is the need to represent ambiguities for indefinite lengths of time in parsing that causes complexity. Sal, however, builds interpretations on-line, and hence ambiguities are resolved locally. For example, most of the ambiguities of the word *can* in example 4.1 were resolved immediately upon seeing the word, because the context was only compatible with the verbal sense of *can*, and ruled out the auxiliary or nominal senses. Those ambiguities which are not resolved by local constraints will often be ruled out by the **Selection Timing Principle**, which prunes less-favored interpretations on-line.

In fact, augmenting Sal by the simple assumption that the interpreter's *working store* is limited in the number of total structures it can maintain, as suggested by Gibson (1991), would insure that the total amount of ambiguity the interpreter can maintain will *always* be limited by a small constant. Although this dissertation does not model processing overload, the overload criteria that Gibson proposes could easily be applied to our model, although it is possible that using realistically large grammars may require even sharper filters than these overload cutoffs.

In each of these cases, placing cognitive constraints on Sal actually simplifies the processing enough to avoid complexity problems.

4.8 Related Architectures

The number of computational models which bear on human sentence processing is enormous. Because it would be impossible to describe all of these models and relate them to Sal in one place, each chapter of the dissertation contains its own related work section. This section, then, will concentrate on surveying related *architectures* for interpretation.

A great many of the models which are not discussed in this chapter emphasize theories of *attachment preferences* or *disambiguation*, and are therefore more productively discussed as selection models in Chapter 7. These include Ford *et al.* (1982), Pritchett (1988), Gibson (1991), Abney (1989), Frazier & Fodor (1978), Shieber (1983), Wilensky & Arens (1980), Schubert (1986), Wilks *et al.* (1985), and Dahlgren & McDowell (1986).

The access mechanisms of many other parsers are discussed in Chapter 5, including Kuno & Oettinger (1962/1986), Aho & Ullman (1972), Kimball (1975), Riesbeck & Schank (1978), Wilensky & Arens (1980), Gershman (1982), Shieber (1985), Pereira & Shieber (1987), Adriaens & Small (1988), van der Linden & Kraaij (1990), Thompson *et al.* (1991), and Gibson (1991).

A number of other models are not discussed altogether, including a number of interesting parsers which serve mainly as implementations of syntactic theories.

This section will briefly survey a number of *classes* of architectures for interpreters. These include semantic analyzers, various parallel architectures, 'compiled-principle' parsers such as some *principle-based parsers*, and finally integrated models whose architectures resemble Sal's.

4.8.1 Semantic Analyzers

Sal's architecture has much in common with the semantic analyzers such as the Yale *conceptual analyzers* (Riesbeck & Schank 1978; Birnbaum & Selfridge 1981) and the *preference semantics* models of Wilks (1975b, 1975c, 1975a). Both of these traditions emphasize the importance of expectations and the use of top-down knowledge in processing.

Wilks models conceptual linguistic knowledge as a set of semantic *templates*. Like frames, templates are semantic structures with gaps and constraints on the fillers of these gaps. Unlike frames, these constraints are expressed as *preferences* rather than as *requirements*, and also unlike frames, templates include syntactic ordering information. An input sentence would be passed through a 'fragmenter' which builds up small structures from the input. A *template matcher* then tries to match templates against these fragments. Wilks's disambiguation mechanism, which was based on choosing the *most coherent interpretation*, is discussed in §7.4.1.

The *conceptual analyzers* of the Yale school (Riesbeck & Schank 1978; Birnbaum & Selfridge 1981; DeJong 1982; Schank *et al.* 1980) also emphasize expectation-driven interpretation. In the most well-specified analyzer in this tradition, ELI, each word which is input to the analyzer will access routines from a dictionary which build conceptual structures. These conceptual structures have gaps, and filling these gaps drives the rest of the processing. This is done by attaching daemons with certain conditions to slots in these structures. When a daemon triggers after seeing some input, it builds structure and fills slots.

Our interpreter's use of valence *holes* as conceptual expectations which can access constructions and guide valence integration, is similar to the use of gaps and templates in the conceptual analysis and preference semantics traditions. Similarly, our use of *coherence* as a selection criterion is similar to Wilks's model.

Differences between the model presented here and the semantic analyzers include the commitment to a theoretically motivated representation of linguistic (including syntactic) knowledge, and to psycholinguistic verification. For example, as §5.3.2 mentions, ELI's use of solely lexically indexed patterns and general lack of higher-level syntactic knowledge makes it difficult if not impossible to represent the complex ordering constraints on adverbs, for example. Similarly, the emphasis in both models on expectations make it difficult for either to access constructions or select interpretations which are not expected, which makes them at odds with the lexical access results of Swinney (1979) and others discussed in §5.3.2.

Later models in the Yale tradition such as DeJong (1982) and Schank *et al.* (1980) concentrated on modeling text *skimming*. Unlike these models, our architecture interprets every word in the input, and thus the interpretation of each sentence involves a large number of lexical and larger grammatical constructions. However these models do make important suggestions concerning the allocation of human attentional capacity to the processing of different words.

4.8.2 Parallel Models

Parallel models of the human sentence interpretation process have been suggested since Fodor *et al.* (1974), but fully explicit models only became common much more recently. A number of architectures for language understanding have used *production-rules* for building interpretations (Riesbeck & Schank 1978; Marcus 1980; Slator & Wilks 1991). Although these production-rules can be designed to operate in parallel, none of these models build parallel structures or interpretations. Even the HEARSAY II model (Erman *et al.* 1980/1981), which did build much of its structure in parallel, did semantic interpretation in a serial fashion — semantic interpretation did not take place until the entire parse tree had been built.

Kurtzman (1985) considers a number of parallel parsing models, reviewing the psychological evidence for each, and concludes that the most favored model is "Immediate Parallel Analysis with strong parallelism". In this model, all possible analyses of the input are built as soon as an ambiguity is detected, and each is updated as further input is processed. A particular analysis is chosen as soon as conceptual and syntactic mechanisms are able to confidently distinguish the most appropriate analysis. Although the details of Kurtzman's model are not specified, the overall architecture is very similar to Sal.

Gorrell (1987) proposes a model like Kurtzman's called the *ranked-parallel* model, which maintains a set of parallel syntactic parses which are *ranked* in terms of simplicity (the smallest number of nodes). Gorrell's architecture is different from the parallel architecture of Sal in two ways. First, the ranked-parallel model builds complete parallel syntactic trees *before* doing any semantic interpretation. Second, while the model builds multiple *syntactic* parse-trees, it only builds a *single* semantic interpretation, based on the highest-ranked parse tree.

Like Gorrell, Boland (1991) presents a model of sentence processing which builds multiple syntactic parse-trees but only a single semantic interpretation. However Boland's model is much more like Sal in building the semantic interpretation at the same time as the syntactic parse and in allowing contextual information to immediately influence the interpretation.

Sal's architecture also resembles Gibson's (1991) parsing model, which consists of a "buffer" and a "stack-set". When words are accessed, each lexical entry is inserted in the buffer along with its "lexical projection", an X-bar maximal category. Thus the buffer will contain one entry for each sense of an ambiguous word. Then each entry in the buffer is attached to each of the parse trees which the system maintains in parallel "stacks" in a "stack-set". A number of selection principles (described in further detail in §7.4.1) instantiate preferences which are used to choose among the parses in the stack-set.

Connectionist models of interpretation, such as Waltz & Pollack (1985), Jain & Waibel (1991), and McClelland *et al.* (1989), are also inherently parallel. In these models, the system does not explicitly consider multiple interpretation, but many of them may receive activation before the system settles on a preferred interpretation.

4.8.3 Compiled-Principle Parsers

A number of recent parsers propose a novel architecture in which the principles of GB syntax are *compiled* into the parser in such a way as to produce the familiar phrase-structure rules or similar knowledge structures. Although these parsers are based on GB theory, they strongly resemble traditional phrase-structure parsers, and they do not produce parses which correspond to the GB analysis of a given sentence. For example, the parser of Correa (1991) uses *attribute grammars* rather than *principles* as its fundamental knowledge structure. In addition, most of these parsers, such as those of Abney (1991), Johnson (1991), Fong (1991), and Correa (1991), compile a number of the GB principles into a new grammar, a *covering grammar*. As Berwick (1991) notes, this new grammar

> is *not* pure X-bar theory — it actually looks more like a conventional context-free rule-based system. . . "

Some of these parsers, such as the *chunk parser* of Abney (1991), use a grammar which bears little if any relation to GB principles at all. *Chunks*, for example, are specifically defined as rewrite rules, and bear a close resemblance to grammatical constructions.

As §2.1 noted, the model presented here is preferable to *compiled-principle* models on the grounds of Occam's razor; the CIG model includes only a single grammar, where the GB model must include two. The fact that the performance grammars used by these parsers resemble construction grammars is additional evidence that a single type of knowledge structure is sufficient.

4.8.4 Integrated Models

One of the earliest integrated models of interpretation which attempted to meet broad-ranging adequacy criteria was the model of Hirst (1986). Hirst's (1986) model included a Marcus-like parser (Paragram), a lexical disambiguation system (Polaroid Words) which attempted to meet psychological adequacy, a semantic interpreter (Absity), and a mechanism for resolving structural ambiguities (the Semantic Enquiry Desk). Hirst's model strongly influenced the design of Sal, but differs in a number of ways. First, where Hirst's model consisted of four separate modules for solving four problems, Sal consists of a single unified mechanism. Second, in embedding a CIG grammar, Sal emphasizes the use of semantic knowledge directly in the grammar, accounting for long-distance dependencies and other phenomena in semantic rather than syntactic ways.

McRoy & Hirst (1990) modifies Hirst's (1986) model with a new *race-based* architecture. This model, inspired by the Sausage Machine of Frazier & Fodor (1978), includes a parser which reads words and incrementally produces a semantic interpretation in two stages. The first stage collects 5 to 7 words into an interpretation fragment, which is then passed on to a second stage and integrated with the complete interpretation. The model simulates parallelism by assigning *time costs* for different attachments or integrations, and choosing the interpretation with the lowest time cost. Structures are combined by the *attachment processor*, which uses a set of *hypothesizers*, specific routines that interact with syntactic and semantic consultant routines to suggest possible attachments.

The system's use of multiple sources of information to suggest possible attachments, and parallel consideration of these attachments, is similar to Sal. The system differs from Sal in its use of time costs as an extremely elegant way of simplifying the selection problem. The system suffers some of the same problems as Hirst (1986), however, in requiring a number of separate modules to solve similar problems. Lexical disambiguation, for example, is handled by the Polaroid Words mechanism, while structural disambiguation is handled by time-costs on different hypothesizers. In addition, the system must include a number of different procedurally-specified hypothesizers to check for thematic expectations, structural expectations, and possible pre- or post-modification, as well as separate consultant routines for phrase-structure-checking and phrase-structure-building. Sal's access mechanism suggests constructions in a more general way by allowing any linguistic evidence to suggest constructions, checking the consistency of the suggestions with the integration algorithm, and Sal's grammar is represented only declaratively.

Cardie & Lehnert's (1991) present a system which resembles Sal in modeling psychological results by using semantic constraints to process wh-clauses, consisting of a semantic interpreter and a mechanism for interpreting embedded clauses. Although their system might be much more robust than Sal, it lacks any representation of larger grammatical constructions, only representing local intraclausal linguistic information. Without a declarative model of linguistic knowledge their model fails to address Representational Adequacy. Also, because their model only allows linguistic structures to be accessed by lexical input, it is unable to account for psycholinguistic results summarized in Figure 1.1 that syntactic, contextual, and frequency information can affect access.

Slator & Wilks (1991) describe an architecture called PREMO (The PREference Machine Organization), which resembles Sal's architecture in many ways. The interpreter maintains a uniform collection of *language objects* which correspond to our interpretation store. As each word of the sentence is input, PREMO creates a new language-object for each sense of the word, and integrates each new lexical-language-object with each language object in the store. PREMO's integration operation is called the **Coalesce** operation. The selection mechanism is based on Preference Semantics (Wilks 1975a; Wilks 1975b).

PREMO differs from Sal particularly in its representation of linguistic knowledge. The system does not use a declarative set of constructions or rules as its grammatical knowledge, but rather a set of *phrase-triggered situation-action rules*. Like the rules of Marcus (1980), these rules are production rules that express which syntactic action to take based on the state of the interpreter and the next word in the input. Because these rules are limited to a small set of five syntactic phrase types, the grammar cannot represent larger, non-lexical, and particularly non-headed constructions. On the other hand, PREMO is able to use the *Longman's Dictionary of Contemporary English (LDOCE)* directly as its lexicon. As such, it is a robust and practical system, unlike Sal, which has a very small grammar and lexicon.

Another difference between PREMO and Sal is that PREMO maintains all possible parses of the inputs, although it only works on the best one at any time. Infelicitous interpretations are never destroyed, but are rather given a very low preference, and hence not pursued.

4.9 Processing a Sentence

This section presents a trace of the interpreter's processing of the sentence *"How can I create disk space?"*. The trace is structured in processing order — each figure follows the previous one temporally. Inside most of the figures are two snapshots — first the access buffer immediately after constructions are copied into it, and then the interpretation store immediately after the access buffer is integrated into it.

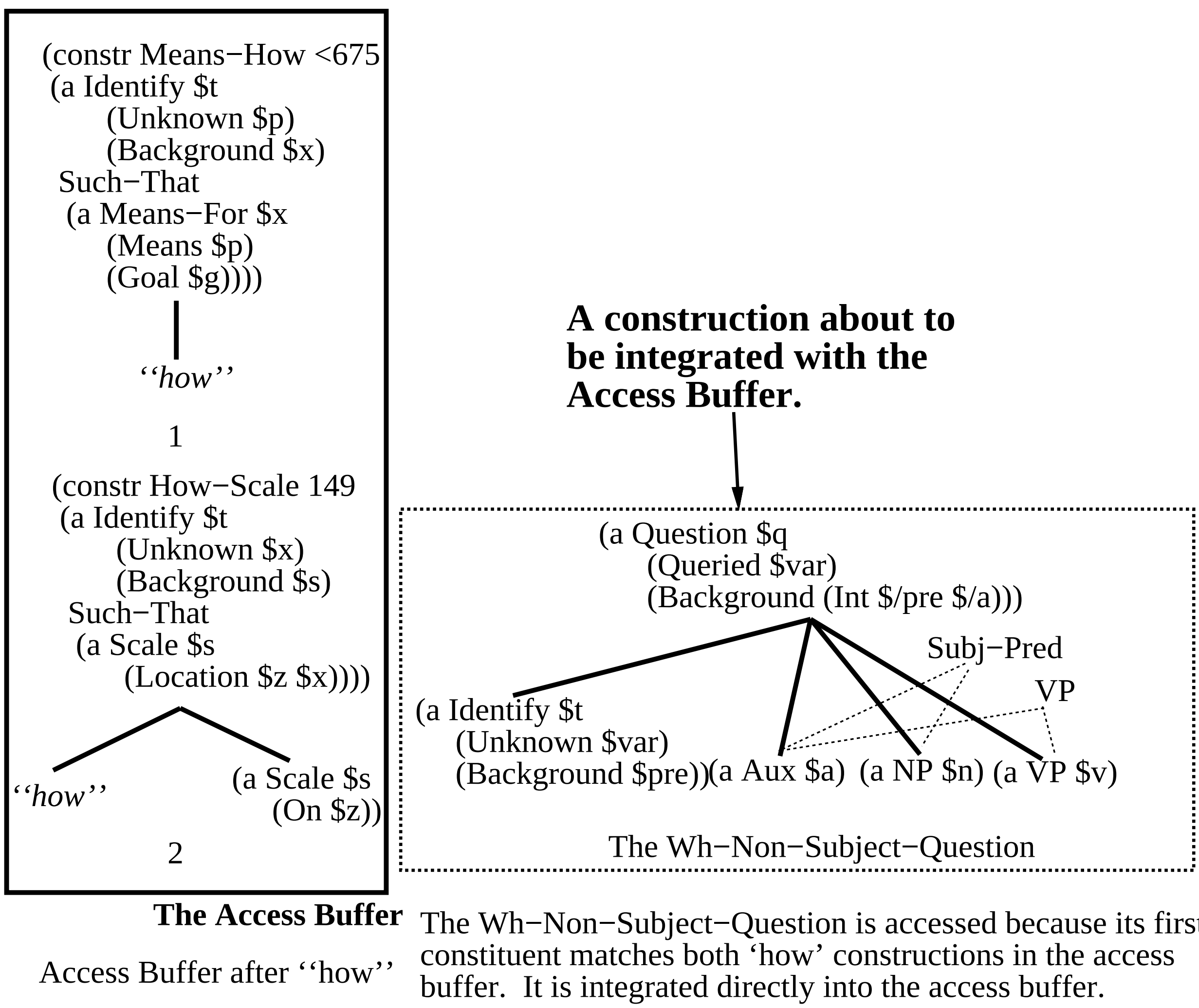


Figure 4.7: The access buffer after "how"

The first figure, Figure 4.7, shows the access buffer after two constructions have been inserted into it, MEANS-HOW and HOW-SCALE. Both constructions were suggested bottom-up by the appearance of the word *how* in the input.

On the right of Figure 4.7 is the WH-NON-SUBJECT-QUESTION construction. This construction is suggested because of the semantics of its left-most constituent. The semantics of this constituent, the **Identify** concept, matches similar semantics in the two constructions in the access buffer (recall that the **Identify** concept characterizes the semantics of all *wh-* elements). Thus the WH-NON-SUBJECT-QUESTION construction is suggested for access by the constructions already in this access buffer. When a construction is suggested this way, it is integrated directly into the buffer, rather than being copied into the buffer (see §4.3 for these details).

Figure 4.8 shows the interpretation store after the two *how* constructions have been integrated with the WH-NON-SUBJECT-QUESTION. The interpretation store now contains four interpretations, although only two are shown in Figure 4.8 for brevity. The two interpretations that are shown are both Wh-Non-Subject-Questions which differ only in how the first constituent has been filled in. The two that are not shown are the original MEANS-HOW and HOW-SCALE constructions, which are also copied into the interpretation store.

The cursors of the two interpretations in Figure 4.8 are in different places. The cursor of the first interpretation has moved to the auxiliary which is the second constituent of the original WH-NON-SUBJECT-QUESTION construction. The cursor of the second interpretation points to the second constituent of the HOW-SCALE construction, which is embedded in the interpretation. Thus the two interpretations place different constraints on the next element to be integrated.

The difference in the semantics of the two interpretations can be seen by a careful examination of the variables in the semantic forms. Note that the **Background** relation of the **Question** in each interpretation is filled by integrating two slashed variables. In the first interpretation, one of these variables is *$x*. *$x* is also bound to the **Background** relation of the constituent MEANS-HOW construction, and thus to the **Means-For** relation.

In the second interpretation, this variable is *$s*. *$s* is also bound to the **Background** relation of the constituent HOW-SCALE construction, and thus to the **Scale** relation.

Figure 4.9 first shows the access buffer after the access of the three lexical constructions which were suggested bottom-up by the appearance of the word *can* in the input. Note that these include the verb *can* (in the sense of "to preserve in a can"), the noun *can* (in the sense of small cylindrical metal container), and the auxiliary *can*. (The verbal sense of *can* meaning "to fire" is not listed here for brevity, since it is processed in the same way as the verbal sense meaning "to preserve in a can").

The right side of Figure 4.9 shows the interpretation store after each of the three senses of *can* in the Access Buffer are integrated with each of the four interpretations in the interpretation store. Recall that this integration process could create up to twelve total interpretations — three constructions times four interpretations.

However, only one interpretation is produced. The other eleven potential interpretations are all ruled out in the integration process in three different ways. First, note that the bottom interpretation in Figure 4.8, the interpretation which included the HOW-SCALE construction, failed to integrate with any of the senses of *can*. This is because the HOW-SCALE construction constrains its second constituent to be a *scale*. None of the senses of *can* in the access buffer includes the **scale** concept. This rules out three of the potential twelve interpretations. The same is true of the bare HOW-SCALE construction which was in the interpretation store (but was not shown in the figure). This rules out three more of the twelve.

Next, the bare MEANS-HOW interpretation is eliminated because it only has one constituent, and thus cannot integrate with further constituents like the *can* constructions.

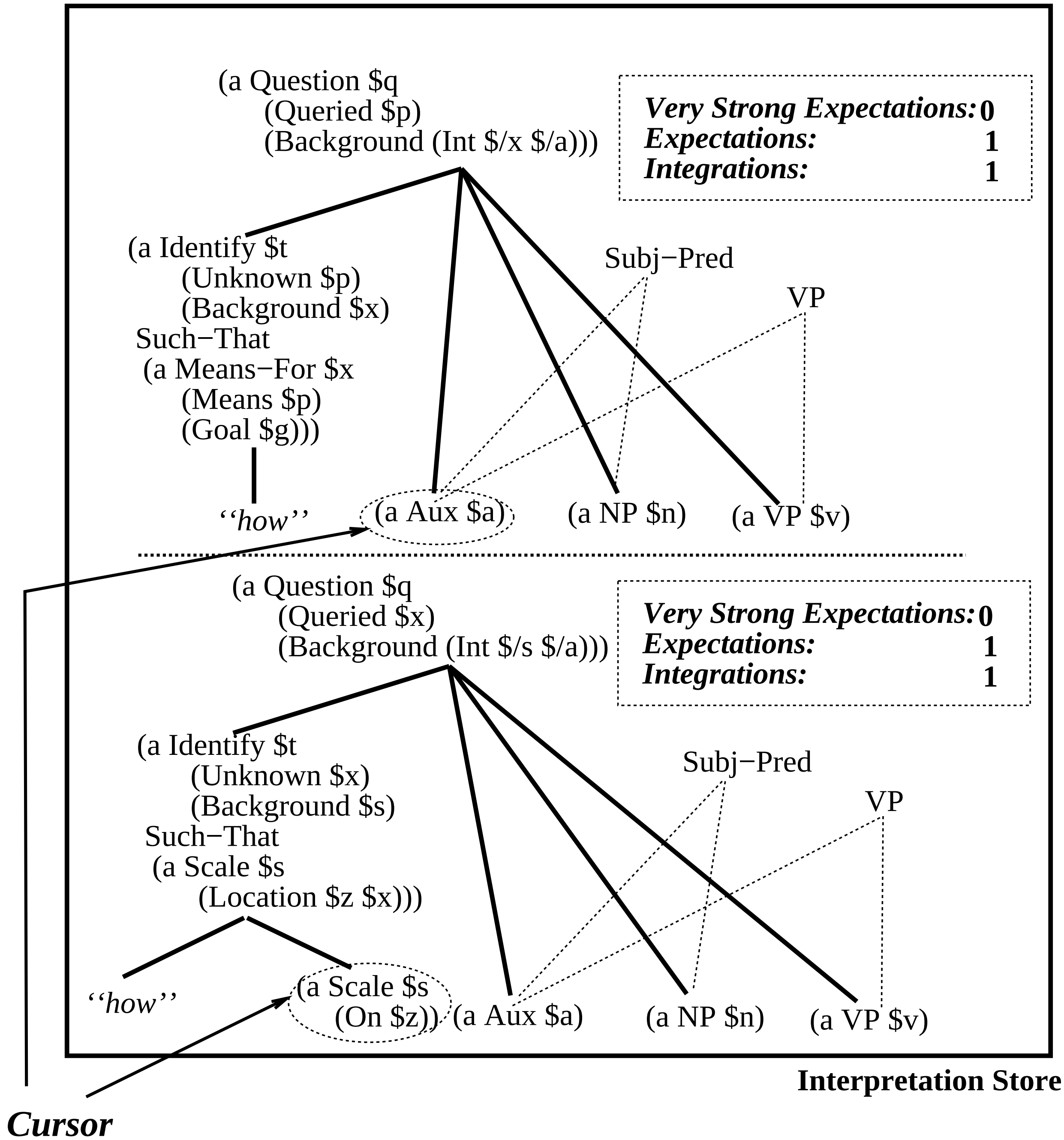


Figure 4.8: After Integrating "how" with the Wh-Non-Subject-Question Construction. Two more interpretations are not shown (see text).

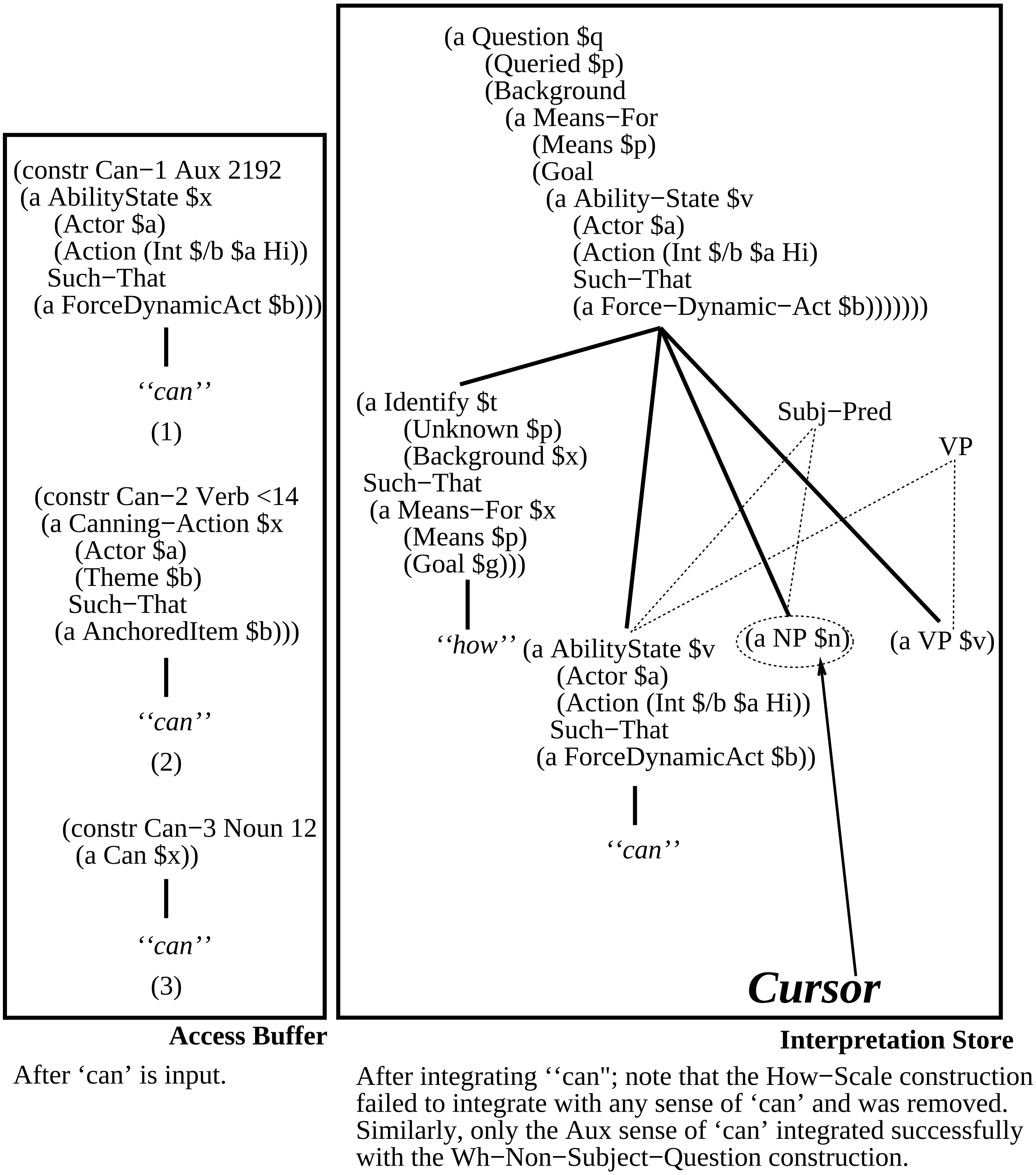


Figure 4.9: Accessing and Integrating “can”

Next, of the three *can* constructions, only the auxiliary CAN is able to integrate with the remaining interpretation, because the interpretation constrains its cursor to be an AUX. Both the nominal and verbal senses of *can* are ruled out.

The semantics of the remaining interpretation in Figure 4.9 builds on the first interpretation in Figure 4.8. Recall that that interpretation had the semantics of Figure 4.10.

```
(a Question $q
  (Queried $*p)
  (Background  (Int $/x $/a)))
Such-That
   (a Identify $t
       (Unknown $p)
       (Background $x)
   (a Means-For $x
       (Means $p)
       (Goal $g)))
```

Figure 4.10: The Semantics of the First Interpretation in Figure 4.8

Figure 4.10 shows the semantics of the first interpretation from Figure 4.8 with the bindings related to the variable *$x* included after the **Such-That** clause. Figure 4.10 might be paraphrased in English as "a question about the means $\$p$ for achieving some goal $\$g$". After the *can* constructions are accessed in Figure 4.9, the semantics of the auxiliary *can* are available to be integrated with the semantics of Figure 4.10. At this point, the interpreter has already built the semantics in Figure 4.10, and it has bound the variable $\$a$ to the semantics of the auxiliary CAN construction. How will the semantics of *can* be integrated with the interpretation?

To answer this question, notice that the semantics in Figure 4.10 specifies that the **Background** for the **Question** is created by integrating the bindings of *$/x* and *$/a*. Because both variables are *slashed*, the integration operation will attempt to find a hole in one of the two semantic structures (the one bound to *$x* or the one bound to *$a*). As the table below shows, the structure bound to *$x* is the **Means-For** concept in Figure 4.10. The only unfilled variable in this structure is the variable *$g* which fills the **Goal** relation.

Variable Bindings	
$x	Means-For
$a	Ability-State
$p	marked as an open variable
$q	Question
$t	Identify
$g	

As the chart shows, all the other variables in Figure 4.10 are already bound, and are not available to the integration operation. The variables *$q*, *$t*, and *$x* are bound because they are in the scope of the operator **a**. As §3.8 discusses, the operator **a** creates an individual, and thus the variable it fills is not an open valence argument. The variable *$p* is previously marked by the WH-NON-SUBJECT-QUESTION construction as being obligatory open inside this construction (see §6.4.3). This leaves only the variable *$g* available for binding. Thus the interpretation in Figure 4.9 shows that the **Goal** relation has been filled with the semantics of the auxiliary *can*, the **Ability-State** concept.

The **Ability-State** concept specifies that a certain **Actor** has the ability to perform a certain **Action**. Note that the **Ability** relation is filled by integrating the **Actor** *$a into* the semantics of the **Action** *$b*. This is how *verbal control* is specified in this grammar.

Figure 4.11 first shows the access buffer after the access of the I construction. This construction is the personal pronoun "I", and is suggested bottom-up by the appearance of *I* in the input.

On the right of Figure 4.11, the I construction is integrated into the interpretation. Note that the **Actor** of the **Ability-State** has become bound to the semantics of I.

Figure 4.12 first shows the access buffer after the verbal CREATE construction is accessed bottom-up after *create* appears in the input. Bottom-up evidence from CREATE construction and top-down evidence from the WH-NON-SUBJECT-QUESTION construction cause the BARE-MONO-TRANS-VP construction to be accessed. As we saw in Figure 4.7, because the BARE-MONO-TRANS-VP construction is suggested by a construction which is still in the access buffer, the new construction is integrated directly into the access buffer. The right side of Figure 4.12 shows the access buffer after this integration. The verbal construction CREATE has integrated with the first constituent of the BARE-MONO-TRANS-VP construction.

Figure 4.13 shows the interpretation store after the verb-phrase containing the verb CREATE from Figure 4.12 has been integrated into the interpretation from Figure 4.11. Note that this verb-phrase has filled the fourth constituent of the original WH-NON-SUBJECT-QUESTION. This constituent was originally constrained to be a verb-phrase, and so the integration is successful.

The semantic result of the integration is to fill in the **Action** relation of the **Ability-State** concept. Note that the **Action** relation is now filled by a **Creation-Action** whose **Creator** is bound to the variable *$i* — in other words the third constituent I.

To see how this semantic integration took place, note in Figure 4.11 that the **Action** relation specified that its filler *$b* must be a **Force-Dynamic-Action** and that the variable *$i* (bound to I) must integrate *into* this action.

Figure 4.14 first shows the access buffer after two constructions are accessed by bottom-up evidence from the appearance of *"disk"* in the input. The two constructions are the lexical construction DISK and the DISK-SPACE construction. The DISK construction suggests the DOUBLE-NP construction, which handles compound nouns. Because it is suggested by bottom-up input, it is integrated directly into the access buffer. The right side of Figure 4.14 thus shows the access buffer after the integration. The DISK construction has been integrated with the DOUBLE-NP construction, leaving three constructions in the access buffer, the original DISK construction, the DISK-SPACE construction, and the DOUBLE-NP construction.

Figure 4.15 shows the (rather complicated) state of the interpretation store after the DISK-SPACE and DOUBLE-NP constructions have been integrated with the interpretation from Figure 4.13.

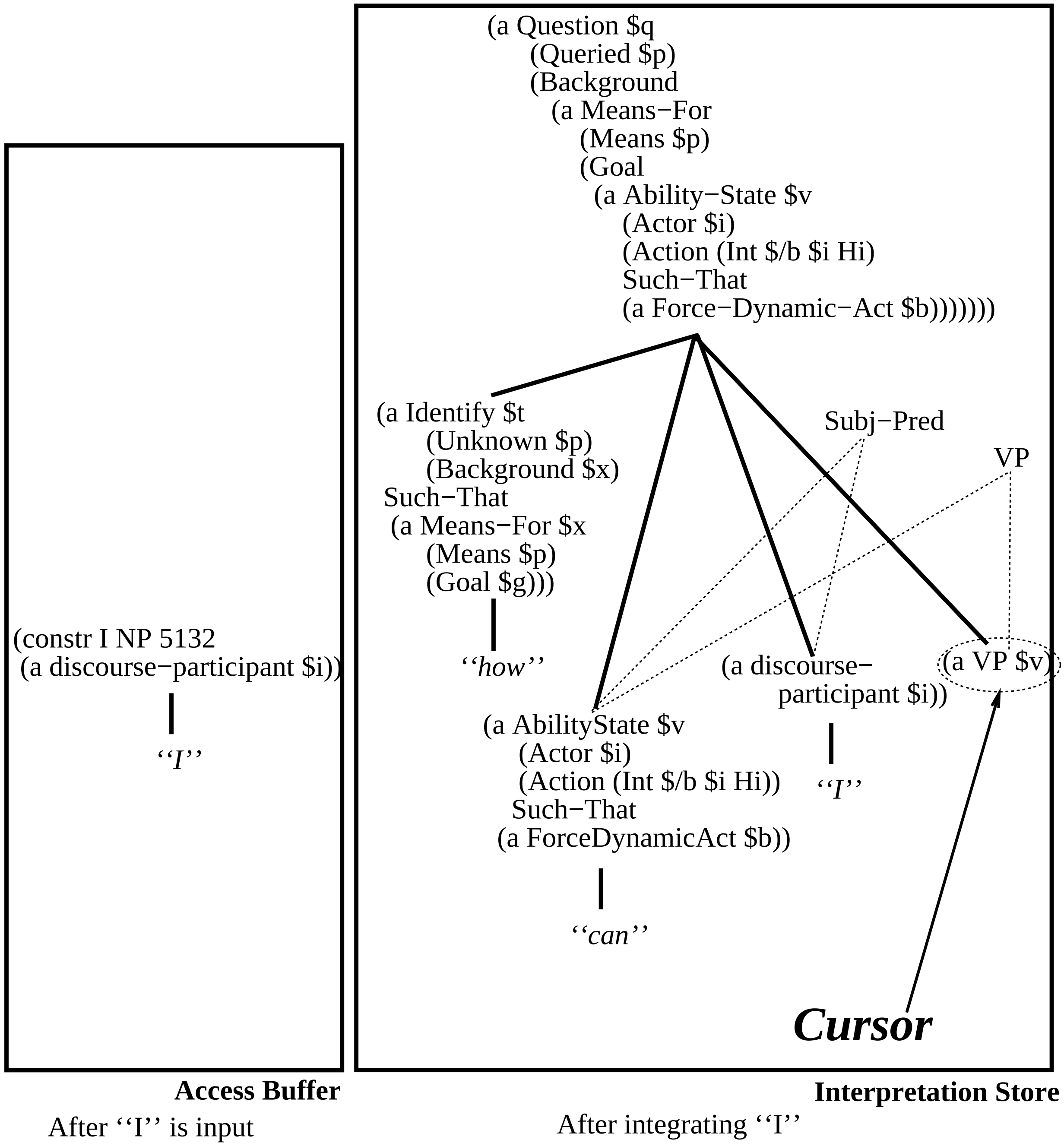


Figure 4.11: Accessing and Integrating "I"

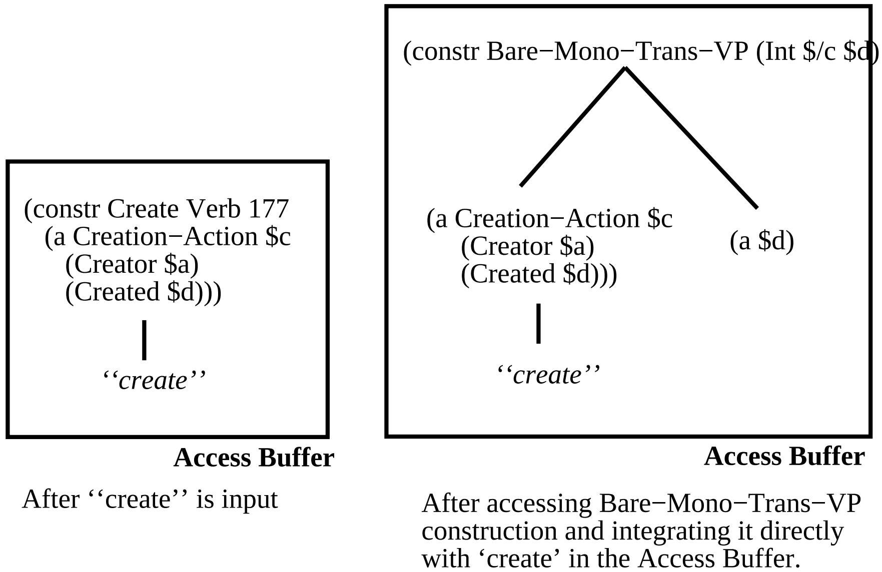


Figure 4.12: Two pictures of the access buffer after "create"

The DISK construction failed to integrate because it did not meet the constraint that required the cursor to be an NP. The interpretation store now contains two interpretations; the one at the top of Figure 4.15 has integrated the DISK-SPACE construction, while the one at the bottom has integrated the DISK construction. The cursor for the top construction points to a constituent which is constrained by the orthographic form "space", while the cursor for the bottom construction is more broadly constrained simply to be a NOUN.

Figure 4.16 shows the access buffer after the lexical construction SPACE is accessed by bottom-up evidence from the input "space".

Figure 4.17 shows the interpretation store after the construction SPACE from the access buffer has been integrated with both interpretations from the interpretation store in Figure 4.15. The integration was successful with both interpretations, and thus both are still present in the interpretation store. But notice the selection scores shown in the upper right of each interpretation. The last integration performed by the top interpretation filled a *very strong expectation*, the expectation for the specific word *space*. According to the Coherence Ranking described in §4.6 and §7.3, filling a very strong expectation gives the interpretation 3 coherence points.

On the other hand, the last integration performed by the bottom interpretation filled a constituent expectation, but not a strong one, and so according to the Coherence ranking it is assigned 1 coherence point. The difference between the two interpretations is 2 points, which is equal to the *selection threshold* σ, and so the bottom interpretation is pruned.

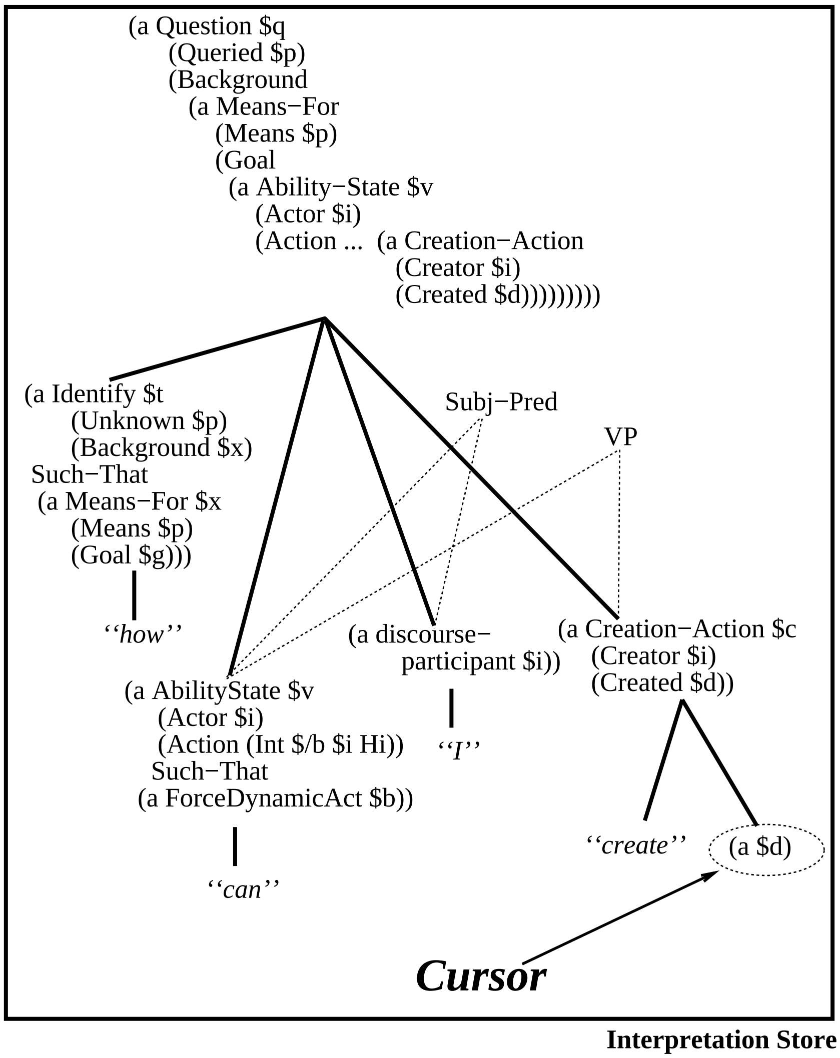


Figure 4.13: The interpretation store after integrating "create"

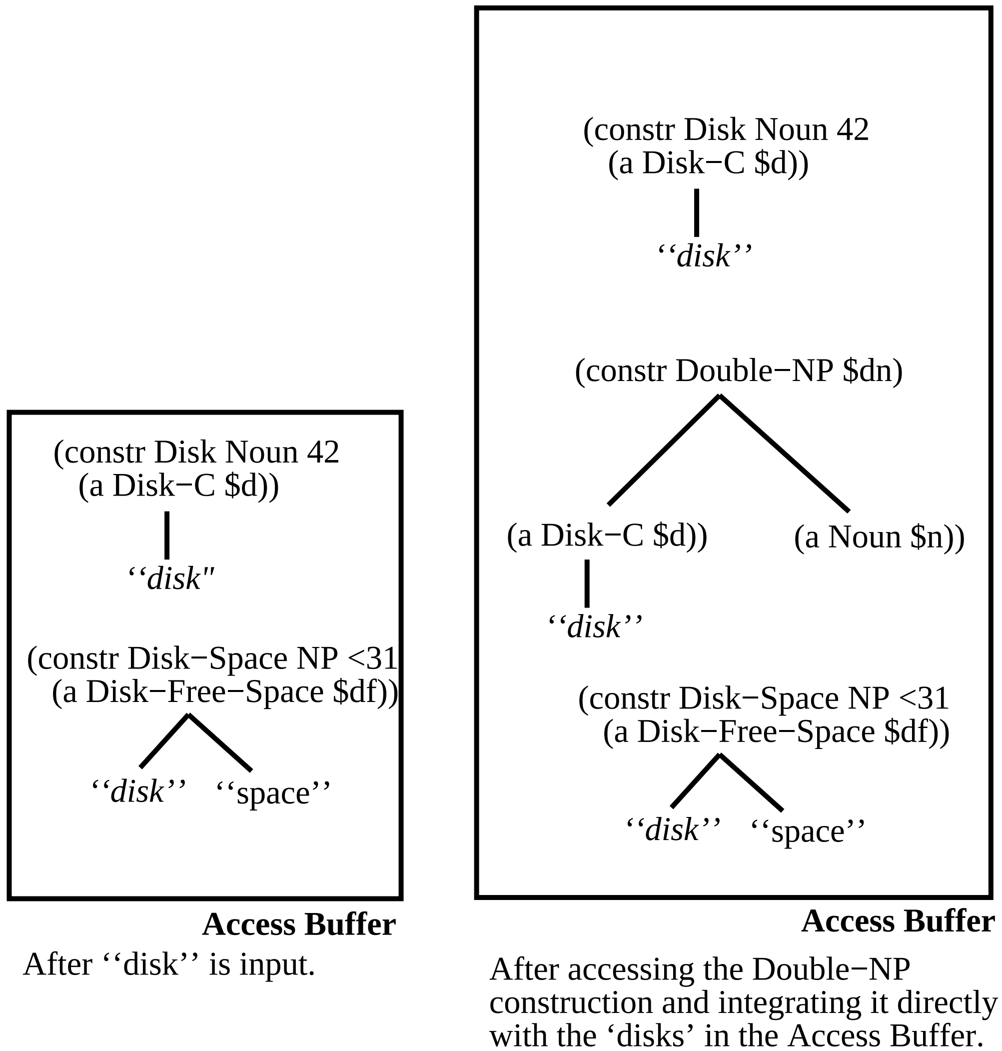
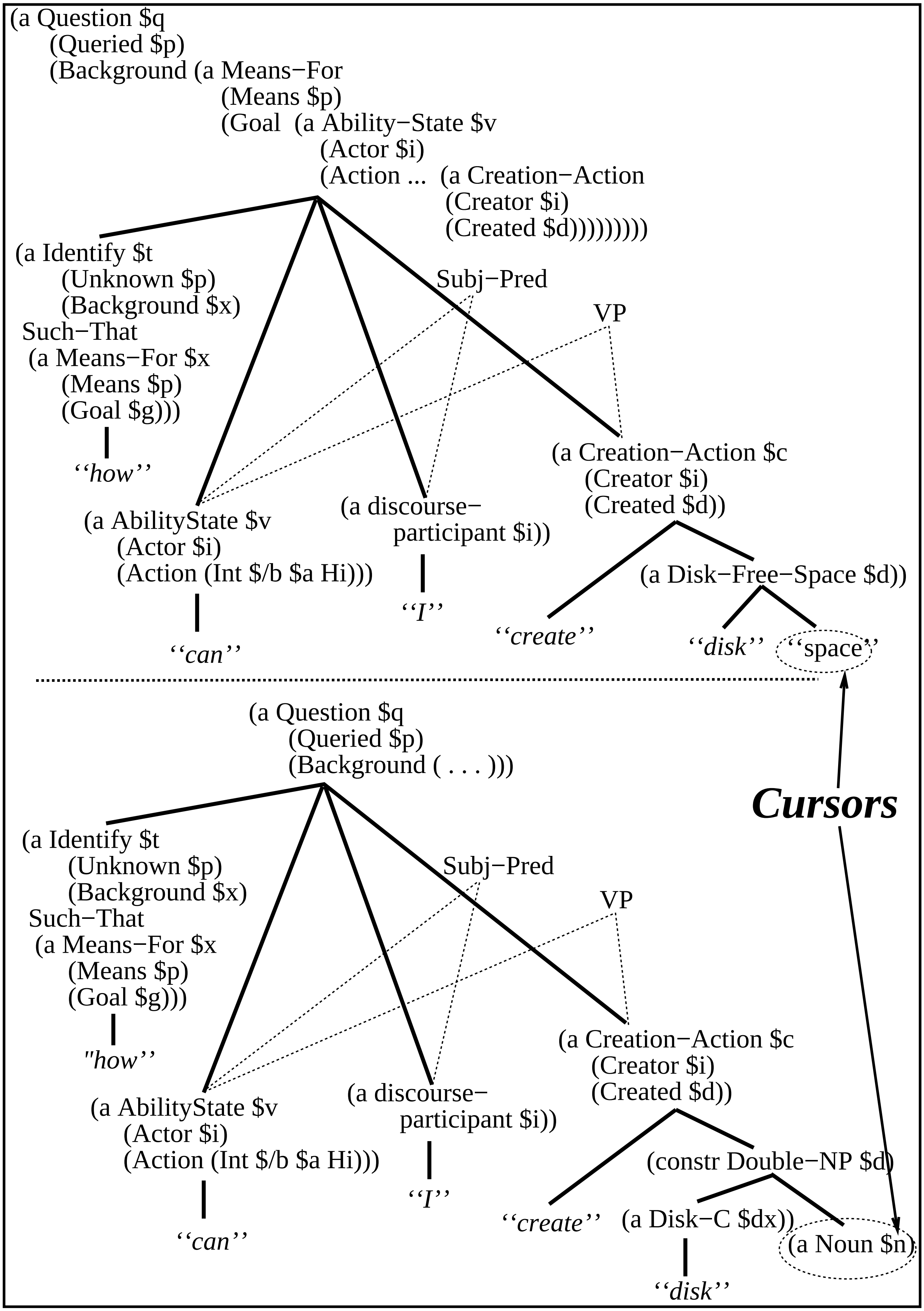


Figure 4.14: The access buffer after “disk”



Interpretation Store

Figure 4.15: interpretation store after integrating two Disk constructions

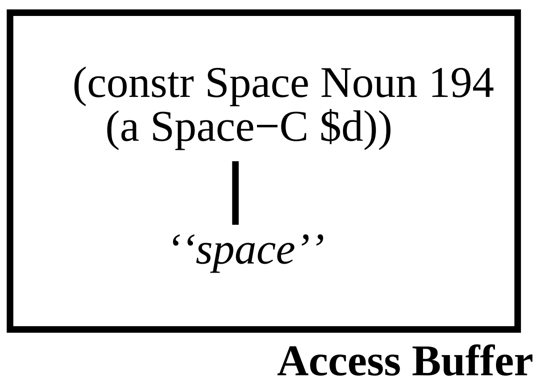
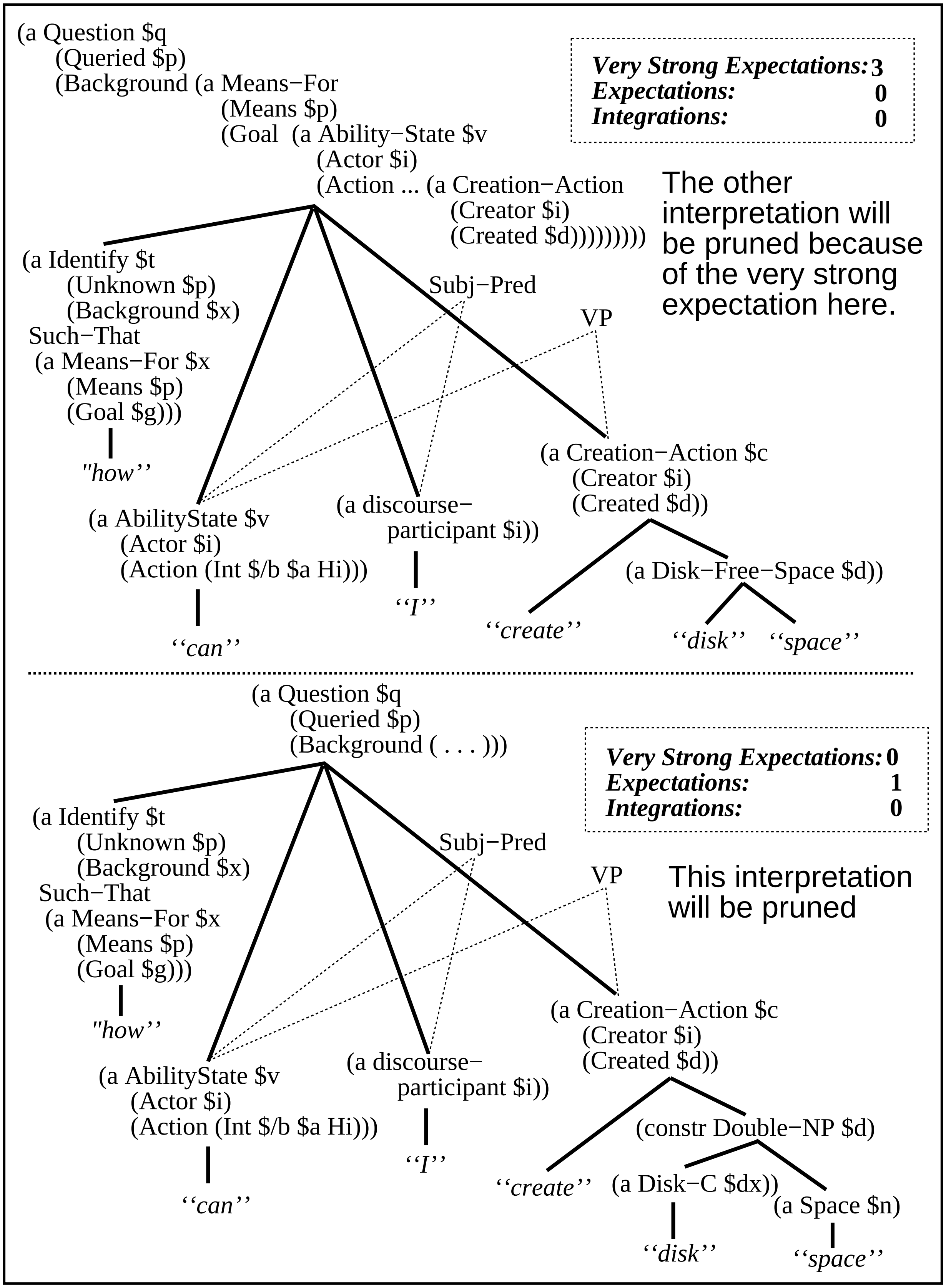


Figure 4.16: The access buffer after “space”



Interpretation Store

Figure 4.17: The interpretation store after integrating “space”

Chapter 5

The Access Theory

A theory of access is an important part of any model of sentence interpretation or parsing. However, although a number of such models have been proposed, research on access has, like the Balkans, tended to be broken up into smaller units and dealt with in an independent and piecemeal way. Psycholinguists have studied *lexical* access extensively, and have studied the access of idioms to a lesser extent, while very little psycholinguistic work has been done on syntactic access. *Syntactic* access has been dealt with frequently in the computational paradigm, by computer scientists and computational linguists who have studied the computational properties of various algorithms for syntactic rule-access in parsing, but with no attempt to model human behavior.

By proposing a single linguistic knowledge base which conflates the lexicon, the syntactic rule-base, idiom dictionaries, and the semantic interpretation rules (the *Grammatical Construction Principle* of Chapter 3), and by using a uniform processing module (the *Uniformity Principle* of Chapter 4), we are able to propose a single access algorithm which accounts for psycholinguistic data and meets computational criteria. We refer to this parallel interactive access mechanism as the *evidential access* model.

Sal's evidential access algorithm is a much more general one than those that have been used in previous parsers or interpreters. Previous models have generally relied on a single kind of information to access rules. This might be *bottom-up* information, as in the shift-reduce parsers of Aho & Ullman (1972), or *top-down* information, as in many Prolog parsers, solely *syntactic* information, as in the left-corner parsers of Pereira & Shieber (1987), Thompson *et al.* (1991), and Gibson (1991), or solely *semantic* or *lexical* information, as in conceptual analyzers like Riesbeck & Schank (1978) or in Cardie & Lehnert (1991) or Lytinen (1991). The evidential access algorithm presented here can use any of these kinds of information, as well as *frequency* information, to suggest grammatical constructions, and thus suggests a more general and knowledge-based approach to the access of linguistic knowledge.

5.1 The Access Algorithm

> **Access Function:** Access a construction whenever the evidence for it passes the access threshold α.

The algorithm can be sketched as follows:
Access Algorithm:

1. Each construction in the grammar has an *activation value*, which is initialized to zero.
2. As the interpreter encounters evidence for a given construction, the activation value of the construction is increased by the number of "access points" corresponding to the new evidence.
3. When the activation value for a construction passes the access threshold α, a copy of the construction is inserted in the access buffer. This point in time is called the "access point",
4. After each access round, the activation value of each construction in the grammar is reset to zero.

The nature of the algorithm mirrors the nature of the interpreter as a whole: access is *uniform*, *parallel*, *on-line* and *interactionist*.

The *uniform* nature of the algorithm follows from the uniform nature of the linguistic knowledge base. Since all linguistic information (i.e., lexical items, idioms, syntactic rules, semantic rules) is represented uniformly as *grammatical constructions*, a single access algorithm can access all this information uniformly. As was mentioned above, lexical access, syntactic rule access, and idiom access are all done by the same algorithm. A construction is accessed by inserting a copy of it into the access buffer.

The next feature of the access function is its *parallel* nature. The algorithm suggests and activates multiple grammatical constructions at a time. Each construction whose activation value is greater than the access threshold α is inserted in the access buffer. For example, in Figure 4.1 in the previous chapter, the string "how" provided evidence for the HOW-SCALE and MEANS-HOW constructions, both of which are inserted in the access buffer in parallel. One of the constructions is lexical, the other has two constituents, and so appears non-lexical.

This activation of multiple constructions simultaneously follows naturally from the **Parallel Principle** of the interpreter, which proposes that the interpreter can maintain parallel interpretations of the input temporarily. As Chapter 4 discussed, there is psycholinguistic evidence supporting parallelism in all varieties of access: *lexical* (Swinney (1979), Tanenhaus *et al.* (1979), and Tyler & Marslen-Wilson (1982)), *idiomatic* (Cacciari & Tabossi (1988)), and *syntactic* (Kurtzman (1985), Gorrell (1987) and (1989), and MacDonald *et al.* (in press)). The evidence for lexical access shows that when an ambiguous input is read, every sense of the ambiguous word is activated.

The next aspect of this algorithm is that it is knowledge-rich and *interactionist*, in using any kind of linguistic information, including top-down or contextual information, to provide evidence for accessing constructions. Earlier access mechanisms were dependent on a fixed set of predetermined syntactic categories or features to suggest rules. In order to allow for the richer information content of grammatical constructions rather than rules, our access algorithm extends these ideas by allowing *any* knowledge that is available to the interpreter to be used to access constructions. Top-down, bottom-up, syntactic, semantic, and lexical knowledge each can be evidence for access of a construction. §5.4 will consider each of these kinds of evidence, and show how they can be used to suggest individual constructions.

Finally, the access algorithm is *on-line*. *On-line* means here that evidence for the access of constructions is accumulated continuously and incrementally. As the interpreter processes constructions which express evidence for other constructions in any of the ways discussed in §5.4 it adds the evidence values to the current state of each construction. When a construction passes the access threshold α, it is copied into the access buffer.

5.1.1 The Access Point

The *access point* is defined as the point in time when the activation of a construction passes the *access threshold* α and the construction is inserted into the access buffer. The fact that a construction is *not* accessed as soon as any evidence for it arrives, but rather waits until enough evidence has accumulated, distinguishes this model from most previous ones; this is an advantage because it captures psycholinguistic results such as those discussed below.

Unlike in these earlier models, the access point of a construction is *not* constant across all constructions, nor is it constant for the same construction in different interpretations. The *access point* thus cannot be a context-independent fixed point in the representation of the construction. Access is context-sensitive: a construction may be accessed earlier in some contexts than others because the context provides more evidence for the construction. This rules out the use of traditional access schemes, where a construction is accessed immediately upon the occurrence of any evidence for it, or more advanced algorithms (like Wilensky & Arens 1980, Cacciari & Tabossi 1988 or van der Linden & Kraaij 1990 discussed below) where access is represented by marking part or all of a construction as the *key* or *indexing clue*.

The *access point* is defined instead as the point at which the construction's activation passes a fixed threshold α. We make the simplifying assumption that this threshold value is the same for all constructions. That is, the interpreter includes a single activation value t, such that when the activation of any construction becomes greater than t, that construction is copied into the access buffer. Thus the access *threshold* α is constant, but the access *point* is not. Different constructions will take different amounts of time to reach the access point because of differences in relative frequency, or in the value of the access cues. Similarly, the same construction would reach the access point differently in different contexts, because the contextual evidence would differ, causing the activation profile to differ.

There are two classes of evidence for this context-dependent *access point* assumed by the evidential access theory. The first class is evidence that access of constructions is not immediate. Swinney & Cutler (1979) showed that some idioms were not accessed immediately after the first content word of the idiom, but rather that it took at least two words to access the idiom. Cacciari & Tabossi (1988) found that for some specially-selected idioms, in absence of context the idiom was not accessed until *after* the last word of the idiom was presented. For lexical constructions, Tyler (1984) and Salasoo & Pisoni (1985) show that the access point for lexical items is approximately 150 ms after word-onset.

The second class of evidence indicates that the access point is variable even for a single construction, in different contexts. Cacciari & Tabossi (1988) showed that access of idioms was faster in the presence of context. Salasoo & Pisoni (1985) showed the same for lexical constructions. Marslen-Wilson *et al.* (1988) showed the negative case — that anomalous contexts can slow down the access point of lexical constructions. Marslen-Wilson *et al.* (1988) showed that the more anomalous the contexts were, the higher the response latencies were.

5.2 The Evidence Combination Function

In line with its interactionist nature, the access function uses a number of different knowledge sources to supply evidence for a construction. These include:

- *Bottom-up syntactic evidence:* For example, the fact that a construction's first constituent matches the contents of the access buffer is evidence for that construction.
- *Bottom-up semantic evidence:* evidence for a construction whose left-most constituent matches the semantic structures of some structure in the access buffer.
- *Top-down syntactic evidence:* when a construction's *constitute* matches the current position of some construction in the interpretation store.
- *Top-down semantic evidence:* when a construction's *constitute* matches the semantics of the current position of some construction in the interpretation store, or matches the semantic expectations of a previously encountered lexical item.
- *Frequency-based evidence:* Constructions are annotated with relative frequencies; higher-frequency constructions are more likely to be suggested.

These various knowledge sources can supply evidence in different ways. Top-down evidence, for example, can be *constituent-based* or *valence-based*. Constituent-based evidence occurs when a construction is part of an interpretation, and one of its constituents has not yet been filled. This unfilled constituent provides evidence for any construction which meets its constraints. If these constraints are semantic ones, then the evidence is top-down *semantic* evidence, if syntactic, then the evidence is top-down *syntactic* evidence. *Valence-based* top-down evidence occurs when the arguments of a predicate are used as evidence for the appearance of a possible argument-filler. When these arguments are constrained semantically, valence-based evidence is top-down *semantic evidence*, otherwise it is top-down *syntactic evidence*.

Proposing an access algorithm which allows multiple kinds of evidence to amass for constructions requires that we choose a *uniform metric* for representing each of these kinds of evidence. We make a simplifying assumption that whatever metric we choose for evaluating evidence, it treat each of these classes of evidence in the same way. Thus bottom-up syntactic evidence values, top-down semantic evidence values and all other evidence values will simply be summed to produce an activation level for a construction.

The difficult question, then, is what metric to use in weighing individual evidence values. Given that each *type* of evidence is weighted equally, the simplest possible combination function might be to also weight each *piece* of evidence equally. That is, we might assign one point to each evidence factor, where a factor might be something like the occurrence of some part of construction in the input. We could assign a constant access point to each construction, say a small integer, and access any construction which receives enough evidence points to pass this threshold.

This metric has a number of advantages, the most obvious being simplicity and operationality. Unfortunately, it has a number of disadvantages. Foremost among these is the fact that individual pieces of evidence differ widely in significance. For example, we would expect that very common words would not be very good evidence for a construction, even if the construction contains these words. This intuition is borne out by Cacciari & Tabossi (1988), which studied idioms in Italian which begin with very common words such as *venire ('come')*, or *andare ('go')*. As we would expect, they found that such idioms are not accessed until after the last word of the idiom was processed. That is, the highly frequent words which began the idiom did not prove a good source of evidence for the idiom, because they provided evidence for so many other constructions as well.

The next factor that our simple metric ignores is the relative frequency of the construction for which evidence is being provided. We would certainly expect that very common construction be suggested more easily and quickly than less frequent ones. Again, this intuition is borne out by a great deal of experimental evidence. A number of studies have shown that high-frequency lexical items have higher initial activation than low-frequency ones (Marslen-Wilson (1990)), are accessed more easily (Tyler 1984 and Zwitserlood 1989), and reach recognition threshold more quickly (Simpson & Burgess 1985 and Salasoo & Pisoni 1985). In effect, frequency evidence acts as the prior probability of a construction, while the other kinds of evidence act as posterior probabilities.

It seems, then, that given a construction e which provides evidence for a possible construction **c**, the construction c ought to receive evidence in direct proportion its own relative frequency, and in inverse proportion to the sum of the frequencies of all the other constructions for which e also provides evidence. That is, if we use E to stand for 'the evidence from construction e for construction c', and x_i to range over all constructions x for which e provides evidence, then:

$$E(e, c) \propto Freq(c)$$

and

$$E(e,c) \propto \frac{1}{\sum_{i=1}^{n} Freq(x_i)}$$

Thus,

$$E(e,c) \propto \frac{Freq(c)}{\sum_{i=1}^{n} Freq(x_i) + Freq(c)}$$

Note that the sum of the frequencies of every possible construction x_i for which a construction c gives evidence must be the frequency of the construction c. That is, if each and only each occurrence of a x_i includes an occurrence of c, there can be no more or less occurrences of c than there are of the x_is. In other words:

$$Freq(e) = \sum_{i=1}^{n} Freq(x_i)$$

In conclusion, we can require of our metric that each piece of evidence for a construction be weighed in direct proportion to the frequency of the construction **p** and in inverse proportion to the frequency of the evidence construction **c**:

(5.1)

$$E(e,c) \propto \frac{Freq(c)}{Freq(e) - Freq(c)}$$

Consider for example, the bottom-up evidence which the input "how" provides for the HOW-SCALE construction. According to Francis & Kučera (1982), "how" has a frequency of 1000 per million, while the HOW-SCALE construction has a frequency of 149 per million. Thus the bottom-up evidence that "how" provides is proportional to 149/(1000-149), or .175. The MEANS-HOW construction, on the other hand, occurs with a frequency somewhat less than 675 (it is not clear exactly how much less, since Francis & Kučera (1982) do not distinguish MEANS-HOW from MANNER-HOW). Thus the bottom-up evidence that "how" provides is proportional to something less than 675/(1000-675) or less than 2.07.

As a simple starting hypothesis, we propose to assign each piece of evidence the weight in "access points" determined by (5.1) above, and to set the *access threshold* α at the value of 0.1 access points. Choosing this low value means that any evidence will be sufficient to access a construction if its frequency is within an order of magnitude of the frequency of the construction.[1]

5.3 Previous Access Models

5.3.1 Syntactic Access Models

Most previous syntactic access mechanisms are quite straightforward. For example a traditional bottom-up parser such as the shift-reduce parsers of Aho & Ullman (1972) (bottom-up parsing

[1]This evidential weighting method is essentially a simple heuristic for approximating the *conditional probability* of the appearance of a construction.

was first suggested in Yngve (1955)) looks at the syntactic categories of the words in the input sentence, and uses this knowledge to suggest rules whose right side matches some handle in the input. This access continues in a recursive way until the structure which has been built reaches the root node.

Top-down parsers (such as the predictive parsers for LL(k) grammars of Aho & Ullman (1972)) begin with the root node of the grammar, and suggest rules whose left-hand side matches some nodes of the parse tree which is being built top-down. Thus if the parse tree contains a verb phrase node, the top-down access algorithm would check the grammar for all the *alternatives* of the verb phrase, (i.e., all rules whose left-hand side is a verb-phrase) and access them.

As a number of researchers have noted (such as Griffiths & Petrick (1965) and Kay (1982)), the only difference between top-down and bottom-up parsers is their access algorithm [2]. Both algorithms use some syntactic information from the phrase-structure which is being built to suggest rules to access. A natural extension of these access mechanisms, then, is to use *both* top-down and bottom-up information to access constructions. Some of the earliest parsers, such as the Harvard Syntactic Analyzer (Kuno & Oettinger 1962/1986), used both kinds of information to access rules. The left-corner parsing algorithm (Aho & Ullman 1972), in which rules are suggested bottom up by their first constituent, and then parsed top down from the other constituents, also uses both kinds of information. This method was proposed as a cognitive model by Kimball (1975) who called it "over-the-top parsing", and is used in a number of systems, including Pereira & Shieber (1987), Thompson *et al.* (1991), and Gibson (1991) extended the idea by increasing the power of the bottom-up suggestion to suggest a construction if its head has appeared (a similar approach was taken by van Noord (1991), who called it a *head-corner parser*).

These approaches to syntactic rule-access could be viewed as methods of *searching* for the correct rules to access, where the search space is the space of possible rules. Bottom-up access amounts to constraining the search by using knowledge of the input. Top-down access amounts to constraining the search by the knowledge of what rules exist in the grammar. Methods which use both top-down and bottom-up information, like the left-corner models discussed above, or the mixed-mode algorithm of Allen (1987), or the connectionist parser of Cottrell (1985), resemble the *version-space* search algorithm proposed for concept learning by Mitchell (1981), which searches for the correct concept by incrementally constraining the space from above and below. In general, the more knowledge which is used to constrain the search, the more likely the search will access exactly the right rules.

The search space of rules is quite different for syntactic parsers, however, than it is for semantic interpreters. All of the syntactic rule access algorithms discussed above were quite simple methods, which were frequently able to compile out much of the access knowledge in advance, because rules were suggested by *syntactic categories*, and the number of syntactic categories in all these systems was quite small. In CIG, however, a construction's constituents may include any set of semantic relations rather than being restricted to a small, finite set of syntactic symbols. Thus these simple access methods used for parsers are insufficient. Many modern linguistic theories have extended the small finite set of non-terminals in a grammar to a larger, potentially infinite set of directed graphs, by allowing constituents to be defined by complex syntactic features. Most of these theories, however, require that the grammar contain a

[2] Although work in parsing tends not to use the term *access* — the term *parsing strategy* (Abney & Johnson 1991) is also used.

"context-free backbone" which is used for parsing. That is, although any constituent may have feature structures of arbitrary complexity, they are required to have a *Cat* attribute whose value is a syntactic category taken from a finite list. In this way the parsers for LFG (Ford *et al.* 1982) and HPSG (Proudian & Pollard 1985) for example, can use the context-free backbone to suggest rules, and use other feature structures to rule them out afterwards. To loosen this dependency on the context-free backbone, Shieber (1985) proposed an algorithm called "restriction", which enables the grammar designer to specify in advance which features the parser should use to suggest rules. Parsers using restriction might use other information besides simple category information to suggest rules. Unfortunately Shieber's method does not allow any way for arbitrary semantic predicates to affect the access process. Thus the evidential access mechanism used in Sal is more general than any of these methods, because it allows any kind of evidence, whether top-down, bottom-up, syntactic, or semantic, to influence the access of constructions.

5.3.2 Semantic Access Models

It is important to note that the use of semantic expectations to guide access was an important contribution of the ELI model (Riesbeck & Schank 1978). However, both ELI and other models in the conceptual analysis tradition (such as the Word Expert Parser (WEP) (Small & Rieger 1982) and (Adriaens & Small 1988)) have also simplified the access problem. In the WEP, each word of the language is modeled as a procedural knowledge source, a "word expert". The word expert contains linguistic and world knowledge about the word necessary to understand it in many contexts. Since all constructions are lexical, there are no higher-level constructions to consider accessing. Although this simplifies the access problem, it means that WEP is unable to represent non-lexical knowledge such as the ordering of adverbials, or knowledge of more general constructions like noun-compounds. Much the same problem holds for ELI, which bases its processing control on semantic expectations set up by words in the sentence which have already been processed. ELI does allow some non-lexical constructions — these are called "traps" and are suggested by the program when the input fails all expectations. Gershman (1982) also notes that access of some post-nominal modifiers must be done by this same "trap" mechanism, while others are handled by routines attached to the individual modifiers. However, the means by which these traps are accessed, and the timing of their access, is not made clear in Riesbeck & Schank (1978) or Gershman (1982).

Riesbeck & Schank (1978) also assumed the *selective access* model of lexical access, in which only the contextually relevant sense of a lexical item is accessed from the lexicon. A number of studies, such as Swinney (1979), Prather & Swinney (1988), Tanenhaus *et al.* (1979), and Seidenberg *et al.* (1982) have presented psycholinguistic evidence which indicates that lexical access is not restricted to the contextually relevant sense.

Riesbeck (1986) proposed that construction access be handled by the same general mechanisms that handle conceptual memory access. This proposal seems quite interesting, but unfortunately the details of the approach are not presented.

5.3.3 Connectionist Access Models

A number of connectionist models of sentence interpretation have been proposed. Like Sal many of these models (such as Waltz & Pollack (1985), Jain & Waibel (1991), and McClelland *et al.* (1989)) are *interactionist*, in allowing semantic and other top-down knowledge to directly affect the access process. The *localist* models (Waltz & Pollack 1985) strongly resemble Sal although they allow a somewhat finer-grained algebra for evaluating evidence for constructions. The *distributed* models (Jain & Waibel 1991; McClelland *et al.* 1989) do not incorporate the same notion of *access* as traditional parsers or interpreters, since rules or constructions are not represented as individual nodes. However, various top-down and semantic influences act as expectations which predict various structural aspects of the interpretation.

Neither of these classes of connectionist models distinguishes between the access and selection theories. There is no discrete access point in these models; structures or features accrue activation in a continuous fashion until one is selected.

5.3.4 Lexical Access Models

Where work on access of more complex structures comes mostly from the computational domain, models of lexical access are mainly psycholinguistic in origin. Simpson (1984) distinguished three classes of lexical access models: *exhaustive access*, *context-dependent access*, and *ordered access*. Simpson's second class, *context-dependent access*, is more perspicuously viewed as two distinct classes — *selective access* and *parallel interactive access*. These models might be arranged according to two variables: *interactive* versus *non-interactive*, and *parallel* versus *serial*, as depicted in Figure 5.1.

The serial models, those on the bottom of the chart, assume that only a single lexical entry is suggested by the access mechanism. Which entry is suggested may be dependent on the context, in the models on the bottom left, or may be solely dependent on relative frequencies, in the model on the bottom right. Researchers in the parallel tradition have argued that the serial models measure the state of the access mechanism *after* the mechanism has settled on a single word.

The top half of the chart lists the parallel models. In the *non-interactive* or *exhaustive access* models in the upper right of the chart, bottom-up stimulus alone determines a set of lexical candidates, and context can only help select the final candidate from among these. The weak interactionist models mentioned in §4.1 assume this form of access, based on results from Swinney (1979), Tanenhaus *et al.* (1979), and the cohort model of Marslen-Wilson (1987). The *Polaroid Words* system of Hirst (1986) implements an exhaustive-access model which then uses semantic constraints to select among candidates.

A slightly modified form of the exhaustive access model, *modified exhaustive access* (Seidenberg *et al.* 1982), allows some associative information from the context to affect lexical access. Seidenberg *et al.* (1982) found that a context including a noun could cause selective access of a semantically-related noun. Cottrell's (1989) connectionist model of lexical disambiguation implements an algorithm which is a generalization of the Seidenberg *et al.* model.

The *ordered-access* model of Hogaboam & Perfetti (1975) serially considers each ambiguous lexical entry, beginning with the most frequent. The search terminates as soon as one entry fits in with the context.

		Interactive +	**Interactive** −
Parallel	+	***Activation−Suppression*** Neill (1989) ***Parallel Interactive Models*** Tabossi (1988) McClelland (1987)	***Exhaustive*** Swinney (1979) Seidenberg et al (1982) Cottrell (1989) ***Cohort*** Marslen−Wilson (1989)
	−	***Selective Access*** Glucksberg et al (1986) Riesbeck & Schank (1978)	***Ordered−Access*** Hogaboam & Perfetti (1975)

Figure 5.1: Previous Models of Lexical Access

In the *selective access* model (Schvaneveldt *et al.* 1976; Glucksberg *et al.* 1986; Riesbeck & Schank 1978), the context completely determines which sense of an ambiguous word is accessed. Although many early non-on-line studies showed support for the model, on-line studies have generally shown effects of multiple-access.

There are a number of problems with these models. In general, these models assume that there is a fixed access point, that is, that a word or cohort is accessed by bottom-up or top-down factors after a fixed time lag, or one which varies only on frequency. The next section summarizes evidence that in fact the access point can vary with the construction and with the context.

The major problem with these models, however, is that they do not extend well to the problem of sentence interpretation. The ordered-access model, for example, assumes that each input will directly index a set of constructions, and then access them serially in order of frequency. While this concept of set-access is quite clear for lexical access, in which the set is the set of homographs, it is difficult to imagine a set of access criteria for grammatical constructions which would return just the right set of constructions for a given phonological input. For example, if the model, upon seeing a verb, accesses all constructions which begin with a verb, it would seem impossible to decide which one is correct immediately.

The final class of models, on the upper left of the chart, are the *interactive* or *context-sensitive access* models, which most resemble Sal's evidential access model. In these models, both context and stimulus can directly affect lexical access. For example, in the *context-sensitive* or *activation-suppression* model of Neill *et al.* (1988) and Neill (1989), multiple meanings of an ambiguous word are accessed in parallel, but the ease of accessing each meaning is a function of its frequency and of the context.

Similarly, in the *interactive activation* framework of McClelland (1987), information at any level of knowledge can affect information at other levels, both above and below — activation flows both bottom-up and top-down. Other interactive models include Simpson & Burgess (1988) and Becker (1980).

In a sense, Sal's access model is a generalization of these parallel interactive models to higher-level structures. While the exact extent to which context and higher-level knowledge can influence access is still debated, it does seem that the larger the structures that are being accessed, the more sense an interactionist architecture makes — since grammatical constructions can be longer than lexical items, the access of a grammatical construction may take longer, thus allowing time for higher-level evidence to take affect.

5.3.5 Previous Models of the Access Point

Finally, a number of previous models have proposed something like an *access point*. The simplest of these models, like the cohort model, assumes that there is a fixed 100–150 ms lag time in lexical uptake, after which access begins (Marslen-Wilson (1987:78)). Proposing that this lexical uptake time is constant for all words effectively means that the cohort model proposes a *fixed* access point, defined in terms of milliseconds after lexical onset. This proposal is incompatible with the evidence of Simpson & Burgess (1985) that the lexical access point is different for different words, as well as the evidence of Salasoo & Pisoni (1985) and Cacciari & Tabossi (1988) that access of even the same construction is faster in the presence of context.

A small number of models of interpretation have also allowed a variable access point. Wilensky & Arens (1980)'s interpreter PHRAN allowed access of a pattern-concept pair (the equivalent of a grammatical construction) to be delayed until more than one constituent of the construction has been seen. Thus constructions like *the big apple*, which occur rarely but begin with common constituents like *the*, are not accessed whenever *the* appears in the input. Some idioms, like *by and large*, which are not lexically headed, were not indexed until the entire idiom had been seen. PHRAN's access model was better than the fixed access lag of the cohort model, but is still fixed for each construction. That is, the PHRAN model, like the cohort model, cannot account for psycholinguistic data indicating that the access point of the same construction is earlier in the presence of context (Salasoo & Pisoni 1985 and Cacciari & Tabossi 1988). The PHRAN model also required the access point for each construction to be determined by the grammar writer. PHRAN's "pattern selection mechanism", the access theory, used a discrimination net to index its pattern-concept pairs, where the grammar-writer was required to specify where each construction was located in the discrimination net. In the access model described in this dissertation, the access *threshold* is fixed for the entire grammar, but the access *point* depends automatically on the construction and the contextual evidence for it, thus eliminating the need for hand-tuning.

More recently, van der Linden & Kraaij (1990) present two algorithms which implement delayed access for idioms. Both algorithms are subsets of the earlier (Wilensky & Arens 1980) model. In the first, the idiom is simply indexed under the first (content) word of the idiom. When that word is recognized, the idiom is suggested. In the second model, which is simpler but interesting because the authors present a connectionist implementation, access is delayed until every word of the idiom is recognized. Both of these models are as inflexible as the cohort model, since both propose a *fixed access point* for all idioms. The first model proposes that *all idioms* are accessed after their first word *in all contexts*; the second that that all idioms are accessed after their *last* word in all contexts.

All of these earlier models thus assume (at best) that the access point is fixed per construction — at worst (in the cohort model or one of the van der Linden & Kraaij (1990) models) a single access point is fixed for the *entire* grammar. None of these models have the ability to use multiple sources of evidence for access, nor allow a context-dependent access point, and thus cannot model variable-access-point results.

5.4 Examples of Access

The next four subsections summarize different kinds of linguistic knowledge that may be used as evidence for a construction.

5.4.1 Bottom-up Syntactic Evidence

Bottom-up syntactic or graphemic evidence is used by all parsers or interpreters. Figure 5.2 below (a part of Figure 4.2 above) shows an example of bottom-up access. After seeing the word "how", the interpreter accessed the two constructions which included that lexical form.

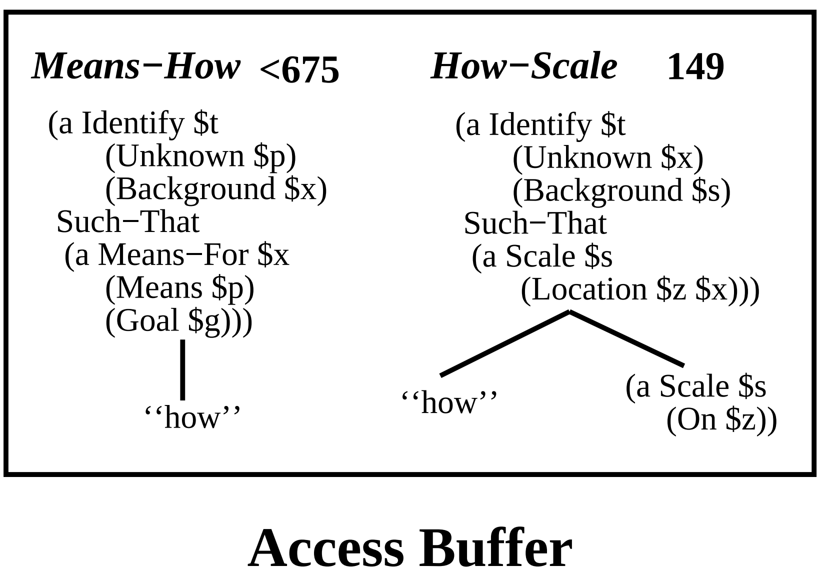


Figure 5.2: The Access Buffer after seeing "how"

The buffer contains two constructions, both accessed because of bottom-up syntactic evidence from the word "how". The first is the MEANS-HOW construction which, as was mentioned above, is concerned with specifying the means or plan by which some goal is accomplished ("How can I get home?"). The second, the HOW-SCALE construction, expresses a question about some scalar properties ("How red is that dress?").

In general, the fact that a construction's first constituent matches the contents of the access buffer will be good evidence for the construction, unless of course the evidential construction is quite rare and the construction is quite common. In each of the cases in Figure 5.2, however, the activation value is greater than the *access threshold* 0.1. The activation of the How-Scale construction is .175, while the activation of the Means-How construction is 2.07, and so both constructions are accessed.

Effects of bottom-up syntactic evidence for access are quite robust in the psycholinguistic literature, as of course one would expect. Thus for example the studies of Swinney (1979) and others cited above show that bottom-up access of lexical constructions occurs even in the absence of context.

5.4.2 Bottom-up Semantic Evidence

In bottom-up semantic access, the semantic structures of some construction in the access buffer provides evidence for a construction whose left-most constituent matches them. For example in Figure 5.3 the semantics of the Means-How construction provide evidence for the Wh-Non-Subject-Question construction.

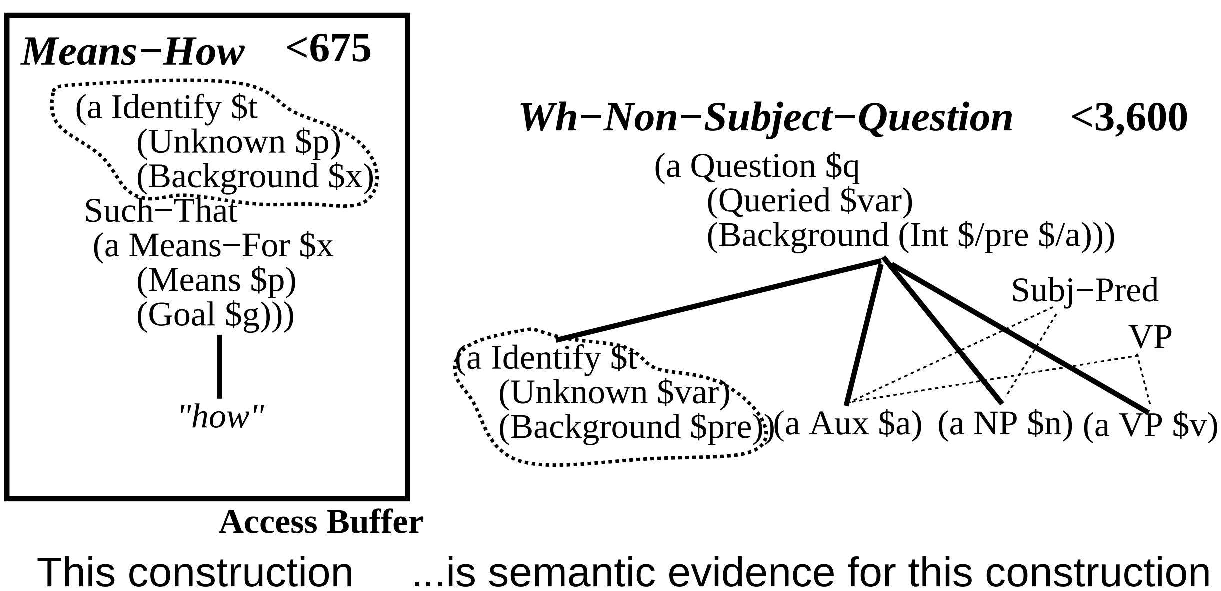


Figure 5.3: Bottom-up Access of the Wh-Non-Subject-Question Construction

Because psycholinguistic results in access have generally been limited to the access of *lexical* structures, and because psychological models have tended to be models of parsing rather than of interpretation it has been difficult to find psychological results which support (or discredit) the notion of bottom-up semantic evidence for access. Recently, however, Gibbs *et al.* (1989) have studied the processing of idioms, and argued for the use of bottom-up semantic evidence in certain idioms. They noted that human processing of a certain class of idioms — those which they called *semantically decomposable* — was much faster than the processing of *semantically non-decomposable* idioms, and than non-idiomatic control sentences. Semantically decomposable idioms are those in which the semantics of the idiom's constituents plays some part in the semantics of the idiom as a whole. For example in the idiom *pop the question, the question* clearly signifies a "marriage proposal", and the verb *pop* the act of uttering it. In a non-decomposable idiom, there is no semantic relation between the meaning of the individual words of the idiom and the meaning of the idiom. For example in the non-decomposable idiom *kick the bucket* there is no relation between *buckets* and dying.

Gibbs *et al.* (1989) proposed that decomposable idioms like *pop the question* or *spill the beans* were accessed when the subjects read the word *pop* or *spill*, because the meanings of these words plays some metaphoric part in the meanings of the entire idioms. That is, the idioms were accessed from bottom-up semantic evidence. Non-decomposable idioms like *kick the bucket* were not accessed until the entire phrase had been seen, because there was no semantic evidence for them.

In order to access idioms from metaphorically related senses in this way, the grammar must include a representation of the conventional metaphors that play a part in the meanings of the idioms. Martin (1990) shows how these metaphors may be represented and learned. Figure 5.4 below shows the representation of the **Spill-the-Beans-As-Reveal-Secret** metaphor that is part of the meaning of the SPILL-THE-BEANS construction, using the notation of Martin (1990), and Figure 5.5 shows the SPILL-THE-BEANS construction which includes this metaphor.

Figure 5.6 shows how the SPILL-THE-BEANS construction would receives bottom-up semantic evidence in the proposed extended model. First, the orthographic input "spill" provides some evidence for the SPILL-THE-BEANS construction, and also provides evidence for the verbal construction SPILL. Next, the *Spilling-Action* concept which is part of the semantics of the SPILL construction in the access buffer provides evidence for the SPILL-THE-BEANS construction, because the SPILL-THE-BEANS construction also contains the *Spilling-Action* concept. This bottom-up semantic evidence thus accumulates in exactly the same way as the **Identify** concept from the MEANS-HOW construction provided evidence for the WH-NON-SUBJECT-QUESTION construction in Figure 5.3 above. The SPILL-THE-BEANS construction thus receives both bottom-up syntactic and bottom-up semantic evidence.

A construction like KICK-THE-BUCKET, which is non-decomposable, only receives bottom-up orthographic input from "kick", but does not receive bottom-up semantic input, since the semantics of KICK are not part of the KICK-THE-BUCKET construction. Allowing the SPILL-THE-BEANS construction to receive evidence from both the input "spill" and the construction SPILL makes the access system different from classic evidential systems, because the orthographic input is in effect providing extra evidence as mediated by the semantics of the SPILL construction.

It is unfortunately not clear from the psycholinguistic data whether the syntactic evidence from "spill" plus the semantic evidence from SPILL is sufficient to access the SPILL-THE-BEANS construction, or whether the syntactic and semantic evidence from "beans" is also necessary. Because Gibbs *et al.* (1989) did not use an on-line measure, the exact access point of the construction is unclear.

The fact that both the literal meaning of spill as well as the meaning of the SPILL-THE-BEANS construction are both accessed, but with varying temporal onsets, is compatible with results from

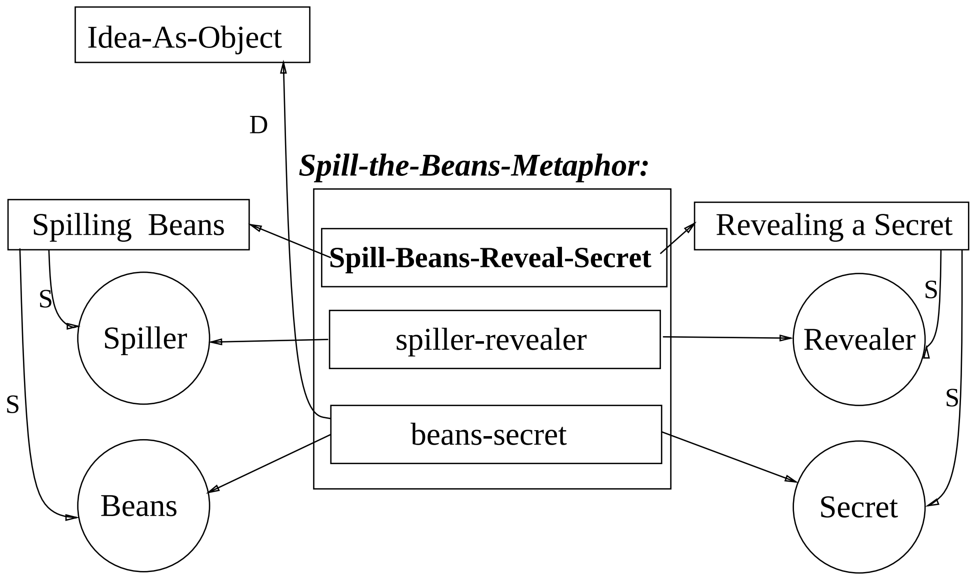


Figure 5.4: The Spill-The-Beans Metaphor (After Martin 1990)

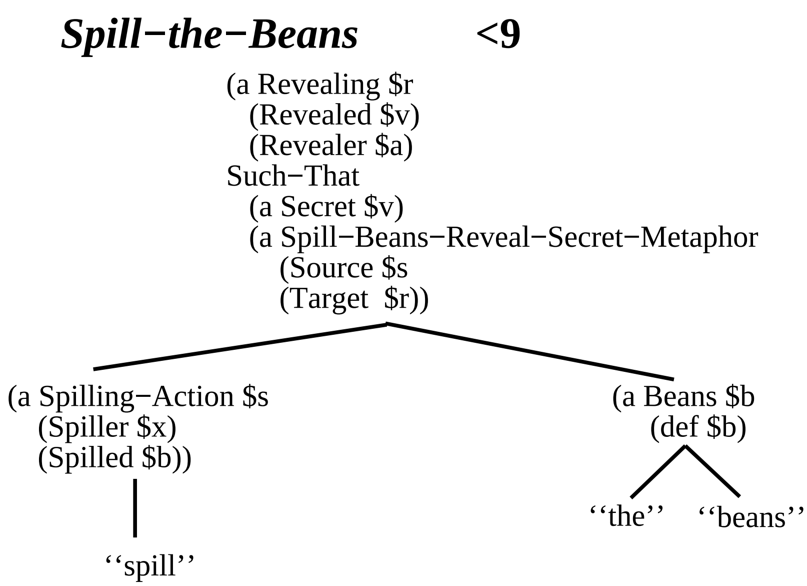


Figure 5.5: The Spill-The-Beans Construction Uses the Spill-the-Beans Metaphor

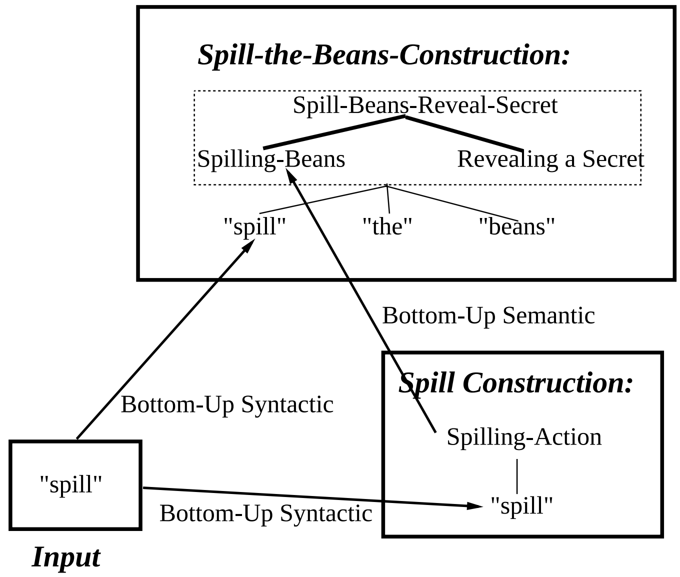


Figure 5.6: The Semantics of "Spill" provides evidence for "Spill-The-Beans"

Cacciari & Tabossi (1988).

5.4.3 Top-Down Syntactic Evidence

As §5.3 discussed, the use of top-down syntactic evidence is one of the historically earliest and also most common access strategies found in models of parsing. Top-down evidence for a construction occurs when its *constitute* (i.e., its left-hand side) matches the current position of some construction in the interpretation store. Figure 5.7 shows an example of top-down evidence. The interpretation store contains a copy of the *Wh-Non-Subject-Question* construction. Because its cursor points at the AUX construction, evidence is provided for that construction. Because AUX is a *weak construction*, the evidence is passed on to the strong constructions which make up the weak AUX construction, and the activation values for these constructions in the grammar rises.

As is the case with bottom-up evidence, top-down evidence may be insufficient to access a construction. One expects this to be true when the top-down evidence does not provide cues that are specific enough to a given construction. For example, constructions which constrain their constituents to be very abstract syntactic categories such as NOUN or VERB (or AUX) do not supply very good evidence for an individual noun or verb. As Figure 1.6 in Chapter 1 showed, the top-down evidence for the AUX construction is insufficient by itself to access any particular auxiliary. For example, although Francis & Kučera (1982) do not specify an frequency for the AUX construction, we estimate it by summing the frequency of the MODAL construction and, conservatively, 75 percent of the DO construction and 20 percent of the BE and HAVE constructions (assuming that these function as main verbs with the complementary percentages). This gives

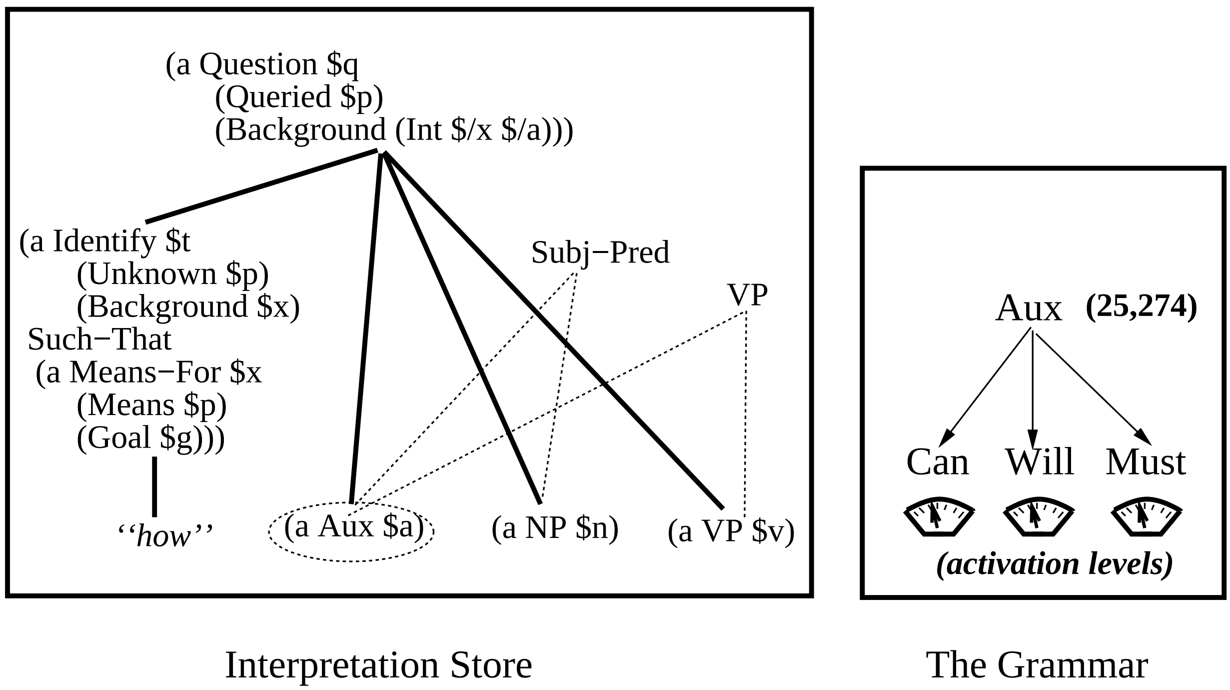


Figure 5.7: Top-down Evidence for the Aux Construction

a frequency for AUX of 25,247. But the frequency of the auxiliary CAN construction is only 1,758. The evidential formula of §5.2 gives an activation for the CAN construction of only (1,758 / (25,247 - 1,758)), or .075, which is below the access threshold of 0.1.

Indeed Tanenhaus & Lucas (1987) note that psycholinguistic evidence of top-down effects are very common in phonology, but much rarer in syntax. They suggest this may be because top-down evidence provides very good cues in phonology, since the conditional probability of a phoneme appearing given a word in which it occurs is 1. (They credit Gary Dell for this observation). The conditional probability of a given construction appearing given a construction which requires it as a constituent is much lower because the constraints are generally specified in terms of abstract constructions like NOUN or VERB. Thus the conditional probability of any *specific* noun appearing is much less than 1. Tanenhaus & Lucas interpret this fact to argue for a difference between the processing of phonology and syntax. Although I agree completely with their evidential analysis, I argue that it is not necessary to propose separate processing mechanisms for phonology and syntax, particularly since there are cases in which top-down access does seem to occur. The uniform evidential access mechanism proposed here can explain both these facts, and still account for the cases of top-down evidence that do occur in the literature.

Two important studies have found evidence for top-down syntactic effects. Wright & Garrett (1984) found that very strong syntactic contexts affected the reaction time for lexical decisions on nouns, verbs, and adjectives. In one experiment, a context ending in a modal verb sharply reduced the time for lexical decision to a verb. Similarly, a context ending in a preposition reduced the time for lexical decision to a noun. Wright and Garrett suggest that their results may be accounted for by proposing that the parser incorporates top-down syntactic expectations for "phrasal heads".

The *evidential access* theory proposed in this dissertation accounts for the Wright & Garrett results in a more general way than specifying expectations for "phrasal heads". This is because *any* open variable which has constructional constraints may act as an expectation and hence evidence for a construction. These variables can be *valence* expectations, such as the expectation from an AUX for a verbal complement that Wright and Garrett found, as well as *constituent* expectations, like the expectation for the AUX construction shown above. Salasoo & Pisoni (1985) also found that top-down evidence, both syntactic and semantic, can cause constructions to be accessed.

5.4.4 Top-Down Semantic Evidence

Like syntactic evidence, top-down semantic evidence can be *constituent-based* or *valence-based*. Consider an example of *valence-based top-down semantic evidence* from the verb *"know"*. This verb is particularly interesting because its arguments have traditionally been assumed to be syntactic rather than semantic. This section shows that the arguments can be expressed semantically, and that they can be used as semantic evidence for the constructions which can fill these arguments.

Consider two of the senses of *"know"*. In the first sense, *"know"* is a stative with two arguments — an animate knower, and some sort of Proposition. This is the *know* of examples (5.2a) or (5.2b) below:

(5.2) a. I know (that) John went to the store.

b. I know (that) my efforts will not go unrewarded.

Figure 5.8 shows the representation of the argument information for this first sense of *know*. Syntactically, these two arguments are expressed as a noun-phrase and a declarative clause, with an optional complementizer "that".

```
(a Knowing $k
    (Knower $a Animate-Agent)
    (Known $b Proposition)
```

Figure 5.8: The semantics of the construction 'know1'

In the second sense, seen in examples (5.3a) and (5.3b) below, the first argument is the same as in the other sense of know — it is constrained to be an Animate Agent. The semantics of the second argument, however, is different; what is known is the (unexpressed) value of the binding for some lambda-expression. Quirk *et al.* (1972) note that this complement of *know* "contains a gap of unknown information, expressed by the wh-element, and its superordinate clause expresses some concern with the closing of that gap, with supplying the missing information."

(5.3) a. I know what color this is.

b. I know what to do.

Figure 5.9 shows the representation of the argument information for this second sense of *know*.

```
(a Knowing $k
    (Knower $a Animate-Agent)
    (Known $b Gapped-Proposition))
```

Figure 5.9: The semantics of the construction ‘know2’

Both these senses of *know* thus have very specific semantic constraints on their arguments. These semantic constraints can be used as evidence to the interpreter to help access the constructions which will instantiate these complements. For example, recall that the first sense of *know* constrained its second argument to be an instance of the **Proposition** concept. This fact provides evidence for the SUBORDINATE-PROPOSITION construction, whose constitute is the **Proposition** concept, and whose syntax builds a finite clause with an optional “that” complementizer, as seen in examples (5.2a) or (5.2b). The second sense of *know*, by constraining its second argument to be an instance of the **Gapped-Proposition** concept, provides evidence for the WH-SUBORDINATE-CLAUSE constructions, which account for the complements in examples (5.3a) and (5.3b) above. (There are three WH-SUBORDINATE-CLAUSE constructions, the WH-SUBJECT-SUBORDINATE-CLAUSE, the WH-OBJECT-SUBORDINATE-CLAUSE, and the WH-INFINITE-SUBORDINATE-CLAUSE, differing in how the *wh*-element is linked to the following verb-phrase).

Figure 5.10 shows the SUBORDINATE-PROPOSITION construction and the WH-OBJECT-SUBORDINATE-CLAUSE, for which the two *know* constructions provide evidence.

The use of the semantic structure of the verb to represent the arguments it allows, rather than subcategorizing the verb syntactically, is discussed in more detail in Chapter 3. There is no evidence bearing on the question of whether a verb’s semantic arguments alone are sufficient to access constructions, although there is extensive evidence that the verb’s *semantic* or *thematic* argument structures are used immediately by the interpreter (including Shapiro *et al.* (1987), Carlson & Tanenhaus (1987), Stowe (1989), Boland *et al.* (1990), Tanenhaus *et al.* (1989), Boland *et al.* (1989), and Kurtzman *et al.* (1991)).

Insufficient Evidence

As with any other kind of evidence, semantic evidence may be insufficient to access a construction. This is especially likely with semantic evidence because semantic structures are more complex than the primitive syntactic categories used by syntactic parsers. For example, consider the various lexical constructions accessed by “how” in Figure 5.2. The second construction, the

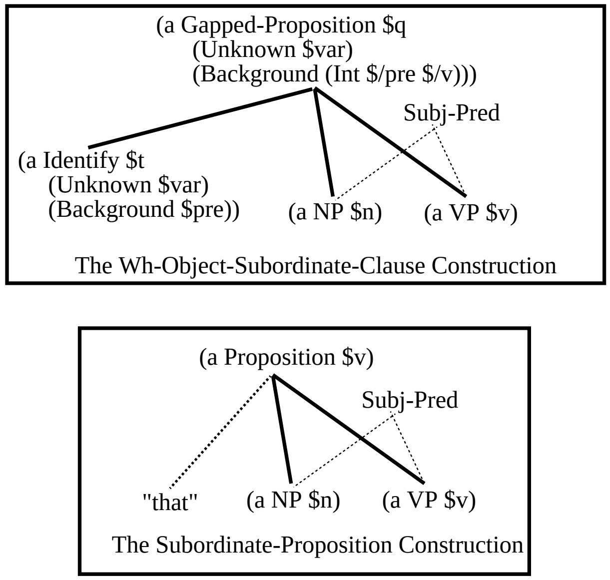


Figure 5.10: The Subordinate-Proposition and the Wh-Object-Subordinate-Clause Constructions

HOW-SCALE construction described in Chapter 3, has two constituents. The first one is the lexical item "how", and the second is described by the semantic predicates **(a Scale $s (On $z))**. After the HOW-SCALE construction is accessed, the access mechanism uses this semantic predicate as evidence to try to access a construction with that semantics. However, this predicate is not good evidence for any particular construction, because it is evidence for so many of them. This is true because there are so many scalar adjectives (e.g., *how big*, *how tall*, *how red*, etc.). As §5.2 shows, the scalar predicate is thus not a good cue for any particular scalar item, and thus no items receive enough activation to pass the access point.

5.5 The Case for Strong Interactionism

The use of semantic information to directly affect the access of constructions makes our access algorithm a *strongly interactionist* one. Obviously all models must allow high-level and contextual information to affect an interpretation; strongly interactionist models are those which allow high-level information to directly cause constructions to be accessed.

Crain & Steedman (1985) and Altmann & Steedman (1988), in defining the terms *strong* and *weak* interaction, note that there are different versions of the strong interaction hypothesis. They consider a situation in which the interpreter is interpreting a sentence beginning with the words "the wife that", in which the presence of multiple possible references for the phrase "the wife" might influence the interpreter in various ways. According to one version,

> the presence in a hearer's discourse representation of several wives predisposes the processor towards complex NP analyses *in general* — that is, not just *the woman that*

> *he was having trouble with* but also *the horse (that was) raced past the barn* Altmann & Steedman (1988:206).

As Crain & Steedman (1985) and Altmann & Steedman (1988) note, this version of the strong interaction hypothesis is unlikely, and it is not the version advanced by this dissertation.

Instead, we propose that the semantics of partial interpretation may help suggest constructions *when the expectation is specific enough to the semantics of the construction*. For example the fact that a verb has a valence argument with specific semantic properties may provide evidence for constructions which instantiate these properties. Crain & Steedman (1985) and Altmann & Steedman (1988) suggest that this version of the strong interaction hypothesis is difficult to distinguish empirically from weak interaction; however it would seem quite possible to study the activation of various grammatical constructions just after the verb has been introduced and before the processor sees any further input.

In introducing the notion of interactionism, Chapter 4 mentioned that the psycholinguistic evidence to date does not conclusively distinguish between the strong and weak interactionist positions. In particular, there seems to be no direct evidence that bears on the question whether contextual or top-down evidence alone can cause a construction to be suggested. There are a number of results, however, suggesting that contextual evidence can speed up or otherwise influence the access process. Wright & Garrett (1984) found that very strong syntactic contexts can speed up the access of nouns, verbs, and adjectives. Salasoo & Pisoni (1985) found that top-down effects, both syntactic and semantic, can cause constructions to be accessed. Cacciari & Tabossi's (1988) study of idiom understanding in context showed that a biasing semantic context sped up access to a given idiom. Marslen-Wilson *et al.* (1988) showed that lexical access was slowed down when a proposed argument to a verb was semantically anomalous. Lexical studies include Oden & Spira (1983)[3], Tabossi (1988), van Petten & Kutas (1988), and Simpson & Kellas (1989).

A number of results have been interpreted to argue against the interactionist position. Most of these, when re-examined, are much more limited in their scope — they argue against the *selective inhibition* or *selective access* position, a position no longer held by most modern theories. The *selective access* position holds that the initial candidate set is strictly limited by context — no constructions can be accessed which are incompatible with contextual information.

For example, the original exhaustive-access studies such as Swinney (1979) and Tanenhaus *et al.* (1979) initially argued that lexical access was independent of contextual influences, because facilitation was found for non-contextually primed senses of words. But as Simpson (1984) and McClelland (1987) showed, in both of these studies the contextually appropriate sense was activated slightly more strongly than the other sense. That is, although access was not *selective*, neither was it blindly *exhaustive*; it does provide extra activation for contextually felicitous candidates. This evidence is thus compatible with the *parallel-interactive*, *context-sensitive*, or *evidential* access models discussed in §5.3.4.

Some studies, however, seem to show incontrovertibly that top-down context does not affect access. A recent study by Zwitserlood (1989), for example, argues that the effects of context are not available until after at least the initial stage of the access phase — about 278 ms into the word.

[3] Although Oden & Spira (1983) tested subjects at least 500 ms after the point of ambiguity, and so it is possible that their results are affected by post-access processing.

One possible conclusion that could be drawn from the conflicting data on interactionism is that access is only sensitive to particularly strong contexts — this is the conclusion reached by McClelland (1987). In addition, it is interesting that none of the studies which argue against interactionism have studied the access of *non-lexical* constructions. It is possible that contextual effects take a certain minimal time to take effect, and thus are particularly apparent with strong contexts or with the larger constructions studied by Cacciari & Tabossi (1988). Clearly more study is needed, particularly with larger structures.

Chapter 6

The Integration Theory

6.1 Introduction

Any theory of interpretation must show how interpretations are built up from (among other things) their component constructions. We call the part of the theory which instantiates this process the *integration theory*. Integration, then, is the process by which the meaning of a construction and its various constituents are incrementally combined into an interpretation for the construction. This incremental interpretation-building has two components: *constituent integration* and *constitute integration*. *Constituent integration* is the process by which a construction's constituent slots are filled by other constructions. In order to fill a constituent slot, a candidate filler must meet the constraints imposed on that slot by the construction. *Constitute integration* is the process by which the semantics of each of these constituents is combined to build an interpretation. Constitute integration may be as simple as linking semantic structures by co-indexing a variable, or may involve more complex combinations of structures.

It is important to note that Sal's integration theory is not intended to be a general solution to the problem of information-combination. As many previous studies have noted, building interpretations requires more than simply combining component meanings. Interpretation requires inference. We divide the interpretation-building process into two components — *grammaticalized combination* and *inferential combination*. The integration operation we define only solves this first class of combinations — those where the grammar specifies how the combination is to be done. Augmenting an interpretation by inferential means such as those of Norvig (1987), Charniak & Goldman (1988), or Hobbs *et al.* (1988) is beyond the scope of this dissertation. This distinction, between grammaticalized interpretation and inferential interpretation, is consistent with a number of experimental results, such as those of Swinney & Osterhout (1990), Murphy (1990)[1], and McKoon & Ratcliff (1990)[2].

In fact, drawing the distinction between *grammatical* and *inferential* combination may help illuminate why certain "inferences" seem to be made on-line, where others are not. For example, a number of researchers, including Garrod & Sanford (1981) and (1990), Singer (1979), and Cotter (1984), have shown that when subjects are given a verb (such as *drive*), they inferred the presence of a role (such as *car*) only if the role was *definitionally* related to the verb. Readers did not infer instruments when the inference required world knowledge, such as inferring the use of a "snow shovel" from a sentence like *Harry cleared the snow from the stairs*. In the model described in this dissertation, inferring **Car** from drive is performed by the integration operation, since the construction DRIVE includes the concept **Car**. However, the integration operation cannot infer "snow shovel" from "clearing snow", since the knowledge that snow-shovels are used to clear snow is not present in any of these constructions.

The following, then, are the kinds of grammaticalized information-combination that are performed by the integration operation:

- Simple combination of constituents in a grammatical construction.
- Combining predicates with their arguments, using semantic and thematic information from the valence description of the predicate.
- Correctly assigning the semantics for the subject of verbs which are *controlled* by other verbs.
- Relating wh-anaphors with their antecedents.

[1]Murphy (1990), for example, showed that integrating noun-modifiers with their head nouns was more difficult when the interpreter had to draw inferences in order to decide exactly how the modifier modified the noun. Simple combinations, in which the information present in the two items was sufficient to make the combination, without external knowledge, were made immediately. More complex combinations took much longer.

[2]By studying only grammatical integrations, and not the inferential combinations, we do not intend to make any claims about modularity or informational encapsulation. We assume, following such models as Hobbs *et al.* (1988) and the *Boots-And-All-Theory* of Hirst (1981), that language understanding necessarily involves many aspects of human cognitive processing. However, in order to circumscribe a more manageable topic, the model presented in this dissertation focuses on linguistic knowledge at the expense of general world knowledge. Thus the fact that the integration algorithm builds certain structures and not others is a function of the knowledge that it is given, not any modularity constraint such as the Modularity Hypothesis of Fodor (1983).

The next section, §6.2, introduces the integration theory and sketches the integration operation. After §6.3 summarizes related models of integration, §6.4 describes the integration operation in detail, and finally §6.5 shows how integration can account for meaning combination in a number of problematic constructions of English.

6.2 A Sketch of the Integration Function

Sal's integration function can be summarized as follows:

> **Integration Function:** An interpretation is built up for each construction
>
> - by applying the *integration* operation
> - in a *constituent-by-constituent* manner
> - as specified by the *constitute* of the construction.

The next three sections will discuss each of these three aspects of the integration function, characterizing it along three dimensions: *which* semantic structures to combine, *how* to combine them, and *when* to combine them.

6.2.1 Which Structures to Combine

The simplest and most common way of determining *which* structures to combine is to specify the combination in a semantic interpretation rule which is linked with a syntactic rule in the grammar, in the style of Montague (1973). When the semantic elements to be combined are all constituents in a single semantic rule, it is simple for the rule to specify exactly which constituents are to be combined and how.

The integration theory uses a derivative of this method, in which the *constitute* of a grammatical construction specifies how the semantics of its constituents are to be integrated. Because a grammatical construction is an abstraction over a complex pairing of meaning and form, there is no need for a distinct semantic rule to accompany the construction, as is employed in Montague's as well as most other theories (Bresnan 1982a; Moore 1989; Pereira & Shieber 1987). Constructions are the only form of linguistic knowledge in our system, and thus it is in the constructions themselves that the instructions for combination are expressed. This choice principle can be simply expressed as follows:

> **Integration Arguments:** Integrate the elements which are specified by the *constitute* of the construction.

For example, in the HOW-SCALE construction defined in §3.4.3 and repeated in Figure 6.1 below, the semantics of the second constituent are integrated with the semantics of the construction because the variable **$s**, which is bound to the assertion in the second constituent, is also bound to part of the **Identify** assertion in the *constitute*.

Specifying which elements to combine becomes more difficult when the semantic elements to be combined are not simply the constituents of a single rule. This occurs with phenomena

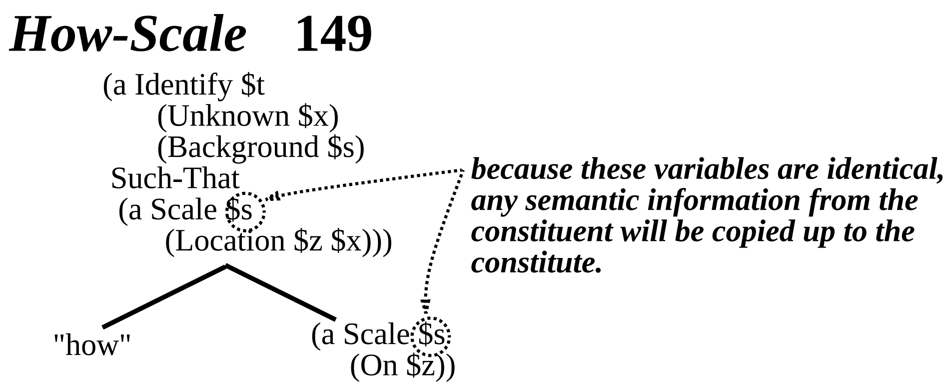


Figure 6.1: The How-Scale Construction Specifies its Integration

like valence, where we would like to integrate the semantics of a predicate and its arguments, and is particularly difficult when the predicate or the argument are related by long-distance dependencies. We see it also with anaphora, where a pronoun must be integrated with its antecedent. Many linguistic frameworks propose special theories which allow them to integrate long-distance dependencies or verbal arguments, such as the *functional uncertainty* of LFG or others.

Our integration theory handles these cases by proposing a more general method to specify that two elements must be combined than simply coindexing their variables or combining their feature structures. This method allows a construction to specify that *$a* and *$b* must be integrated, where in fact *$a* should be integrated with some variable *inside* the structure which fills *$b*. The integration process will attempt to find an appropriate semantic gap (called a *hole*) in *$b* to bind to *$a*. The VERB-PHRASE construction, for example, specifies that the complement of the verb must be integrated with some hole *inside* the semantics of the verb.

This extension to the simple semantic-interpretation-rule method requires that the integration operation be more powerful than simple operations such as unification or functional-application, so that it can decide exactly which elements are to be combined. This extra power that *valence-integration* requires, and the details of valence-integration, are discussed in detail in §6.4.3.

An important feature of this algorithm is that it does not treat long-distance dependencies as the result of *movement*, mediated by some coindexed empty category. Long-distance dependencies are resolved in the semantic domain, and are handled in the same way as other kinds of integration (see §6.5).

6.2.2 How to Combine Structures

Once an integration theory has determined *which* constructions to integrate, it must decide *how* they are to be integrated. The kind of integration theory that we will define here resembles the Universal Grammar of Montague (1973), as well as the unification-based semantic interpretation theories of Pereira & Shieber (1987) and Moore (1989). Like these theories, our integration theory includes an algorithm which, given a set of semantic structures, produces a combination of their meanings. Unlike in these theories, strict compositionality is not essential to integration — the interpretation of a construction may be augmented by combination with contextual or world knowledge. Indeed, as Chapter 3 showed, grammatical constructions themselves are defined specifically when there is non-compositionality — i.e., some element of meaning not predictable from the constituents. At the risk of confusion, the operator itself is called *integration*, and so the combination principle is expressed as follows:

> **Integration Method:** Apply the *integration operation* to each of the specified elements to produce an interpretation.

As §6.2.1 suggested, the integration operation is a somewhat more intelligent one than unification or functional composition. We consider here four ways in which integration extends the unification operation:

- The integration operation is defined over the semantic language defined in §3.8 rather than the *feature-structures* used by *feature unification*. This allows the interpreter to use the same semantic language to specify constructions as it uses build final interpretations, without requiring translation in and out of feature structures.

- The integration operation distinguishes *constraints* on constituents or on valence arguments from *fillers* of constituents or valence arguments. This extension solves a traditional problem with unification-grammars. In pure unification grammars, there is no way to know when the argument of a verb has been filled, because unification does not distinguish between *constraints* on an argument and a *filler* of an argument — both are represented as feature structures. Integration solves this problem by distinguishing constraints from fillers with the *marking* algorithm described in §6.4.3.

- Because the integration operation is defined for a specific representation language, it can use information about the representation language to decide if structures should integrate. For example, if a construction constrains one of its constituents to be a *weak construction* like DETERMINER, this constituent will integrate successfully with a strong construction like THE, because DETERMINER abstracts over THE. Examples of this are presented in §6.4.2.

- The integration operation is augmented by a *slash* operator, which allows it to join semantic structures by embedding one inside another. This is accomplished by finding a *hole* inside one structure (the *matrix*), and binding this hole to the other structure (the *filler*). This approach resembles the unification-based formalisms of Pereira & Shieber (1987) and Moore (1989), which extend unification by borrowing the idea of *lambda-abstraction* and *functional application* from categorial grammar (Adjukiewicz 1935/1967). The slash extension is more complex than function-application because fillers must meet the semantic constraints which are posted on holes. The difference between integration and functional application are discussed in §6.3.

The integration algorithm is discussed in detail in §6.4.

6.2.3 When to Combine Structures

The final question that must be addressed for an integration model is *when* to integrate. Answering this question traditionally means choosing a *granularity* for the interaction between syntax and semantics, deciding how often semantic interpretation rules should be activated. Unlike CIG, most models distinguish syntactic and semantic rules, and thus the structure-building performed by each can be quite distinct. Following a great deal of psycholinguistic evidence which argues that integration must be *incremental*, the integration algorithm chooses the most fine-grained integration timing which is possible:

Integration Timing: Perform integration *constituent-by-constituent*.

Consider first the way that other models have chosen to time semantic integration. A great number of early models have chosen to perform semantic interpretation *after* an entire sentence or clause has been processed. Models which made this assumption of a very broad granularity for syntax-semantics interaction include Fodor *et al.* (1974), Erman *et al.* (1980/1981), Woods (1977), and McCord (1982) and (1990).

Systems which interleave syntactic and semantic processing at a somewhat finer granularity include Marcus (1980) and Winograd (1972). The semantic knowledge of Winograd's (1972) SHRDLU consisted of a large number of procedures which examine the syntactic parse of the input and build up a PLANNER program to answer the question. These semantic routines were called at various times in the syntactic parse, constituting a medium-grained interaction. For example, the NOUN GROUP specialist was called first after the head noun of a noun phrase, and then after the modifiers. Similarly, Marcus's (1980) Parsifal was augmented with a set of *attachment monitors* as part of a Case Frame Interpreter designed to produce a case-theoretic interpretation of a sentence. Although these monitors are triggered by a number of possible events in the parse, they do not trigger until *after* a verb has been parsed, and like SHRDLU's routines, generally trigger only at the *end* of noun phrases.

Most recent models assume that semantic integration take place at a finer granularity than these models, assuming that semantic integration takes place at every *reduction* — that is, *after* a construction or rule has been completed. This is the *rule-to-rule* method defined by Bach (1976). Models which use this approach include Hendrix (1978/1986); Pereira & Warren (1980); Schubert & Pelletier (1982); Altmann & Steedman (1988); Steedman (1989) and Haddock (1989) (although the last three models effectively redesign the rule-to-rule approach to achieve a finer granularity, as will be discussed below). A number of researchers, including Altmann & Steedman (1988); Steedman (1989); Haddock (1989) and Stabler (1991) have noted that if integration only takes place *after* a reduction, it cannot be as incremental as psycholinguistic evidence suggests

In the model presented in this dissertation, the granularity of syntactic-semantic interaction is more fine-grained than any of the models discussed above. We call this granularity *constituent-by-constituent*. A partial interpretation for each construction is constructed as soon as the construction is suggested by the access mechanism, and as each constituent of any construction is proposed, its semantics are integrated with the *constitute* of the construction. Thus an interpretation is available as soon as the smallest sub-constituent is integrated. Indeed, because syntactic and semantic constraints are represented uniformly in grammatical constructions, it would be impossible for syntactic and semantic structure-building to be disjoint.

There is a great amount of psycholinguistic evidence for this fine-grained, on-line nature of interpretation building, including evidence from comprehension (Marslen-Wilson 1975; Potter & Faulconer 1979), lexical disambiguation (Swinney 1979; Tanenhaus *et al.* 1979; Tyler & Marslen-Wilson 1982; Marslen-Wilson *et al.* 1988), pronominal anaphora resolution (Garrod & Sanford 1991; Swinney & Osterhout 1990), verbal control (Boland *et al.* 1990; Tanenhaus *et al.* 1989), and gap filling (Crain & Fodor 1985; Stowe 1986; Carlson & Tanenhaus 1987; Garnsey *et al.* 1989; Kurtzman *et al.* 1991). Potter & Faulconer (1979) present quite specific results showing that the integration of the two constituents of the ADJECTIVE-NOUN construction is done immediately; they found that the interpretation for an adjective-noun pair was available immediately at the offset of the noun.

A number of other recent models propose on-line, incremental integration models. The reading model of Just & Carpenter (1980), for example, assumed that some integrations would be immediate. The HPSG parser of Proudian & Pollard (1985) allowed constituent-by-constituent interpretation, but only after the *head* had been found in the input — in cases where the head appears late in the input, the granularity was more like *rule-to-rule*. The integration algorithm in RUS (Bobrow & Webber 1980) seems to be more fine than rule-to-rule. RUS was an ATN parser linked with a semantic interpreter, PSI-KLONE. At certain arcs (all arcs?) of the ATN, the parser proposes functional relation between syntactic constituents, and the semantic interpreter responds by accepting or rejecting. It is difficult to tell from the paper whether this syntax-semantic interaction happened after *every* constituent or merely most of them.

The *categorial grammar* proposals of Altmann & Steedman (1988); Steedman (1989), Haddock (1989), and Hausser (1986) redefine the rules of categorial grammar to produce left-branching structures so that the rule-to-rule method will produce the same results as the constituent-by-constituent approach. Steedman and Haddock both claim that categorial grammar is thus more amenable to incremental interpretation than other models, since it can produce an incremental interpretation while maintaining the advantage of rule-to-rule parsing in keeping a clean relation between syntactic and semantic processing. The constituent-by-constituent method also has both of these advantages, since integration is incremental and the production of an interpretation is directly tied to the construction which licenses it. Since the semantics of a construction (i.e., its constitute) is expressed as a set of assertions with variables, a partial interpretation is available as soon as the construction is accessed, and since each word of the input will be a constituent of some construction, constituent-by-constituent integration implies that an interpretation can be incrementally augmented as each word is processed. This allows the constituent-by-constituent method to avoid what Stabler (1991) has called the *Pedestrian's Paradox*; the pedestrian's paradox is the assumption that a semantic interpretation cannot be assigned until after a rule has been completed and reduced.

Although integration takes place incrementally, a number of experiments have shown that some parts of the integration process may occur only at clause or sentence boundaries, acting to integrate the sentence with previous parts of the text. Because the interpretation model discussed in this dissertation does not focus on inter-sentential processing, and because the integration algorithm only models *grammaticalized* combinations, the integration algorithm does not model these slower, more powerful end-of-sentence processes.

6.3 Previous Integration Models

6.3.1 Information-Combining Formalisms

Formalisms which extend unification-like approaches from the syntactic to the semantic domain, such as Moore (1989) and Pereira & Shieber (1987), have used the *lambda-calculus* to represent the functional nature of the partial information structures, and *functional application* to combine these structures. For example if the verb *halt* were represented as $\lambda x halts(x)$, then applying this function to an element like *SHRDLU* would produce the form *halts(SHRDLU)*.

The integration theory discussed here might be viewed as using *implicit lambdas* for every partial information structure. All unfilled variables (i.e., unmarked variables, see §6.4.3) are considered *open* by the valence-integration algorithm, and thus act as if the information structure was in the scope of the appropriate lambda. The valence-integration algorithm ignores variables which are already filled, which thus act as if they were *not* in the scope of a lambda.

6.3.2 Valence- and Gap-Filling Formalisms

Both of the unification-based formalisms mentioned above, (Moore 1989 and Pereira & Shieber 1987) propose similar mechanisms for representing and integrating filler-gap dependencies, the *argument stacks* of Moore (1989), and the *gap-threading* of Pereira & Shieber (1987). The *gap-threading* algorithm, for example, propagates gap information in two directions — top-down information from constructions which require a gap to occur, and bottom-up information from the lexical gap-insertion rules which indicate that a gap exists.

Because filler-gap integration is done *semantically* rather than *syntactically* in CIG, there is no need for gap-threading. A construction specifies that a hole is required by binding a filler to a *slash*-variable. When the constituent which instantiates the slash-variable is found, the integration algorithm finds a hole inside it to bind the filler. There is no need for lexical-insertion rules which add empty-categories to the phrase-structure tree, and hence no need to back-propagate gap-location.

Like CIG, the filler-gap relation in LFG is also expressed in non-phrase-structure terms. Kaplan & Zaenen (1989) and Kaplan & Maxwell (1988) proposed that a long-distance-antecedent is linked directly with the *functional* structure of a predicate. The *functional* or *f-structure* level of LFG is defined in terms of grammatical relations like TOPIC, OBJ, and COMP.

Linked with this proposal for a functionally-based account of long-distance dependencies is a representational mechanism called *functional uncertainty*. Functional uncertainty allows a kind of abstraction in the equations which specify how the fillers of different functions are related to each other. For example, consider the topicalized sentences (6.1) and (6.2) from Kaplan & Zaenen (1989) (their (25) and (26)):

(6.1) Mary John telephoned yesterday.

(6.2) Mary John claimed that Bill telephoned yesterday.

In 6.1, the appropriate LFG equation relating the topicalized element and its subcategorizing predicate is $(\uparrow TOPIC) = (\uparrow OBJ)$, indicating that *Mary* is the object of the verb *telephoned*. In 6.2, the appropriate equation is $(\uparrow TOPIC) = (\uparrow COMP\ OBJ)$, indicating that *Mary* is the object of the *complement* of the verb *telephoned*. In general, then, the equation for constructions of this sort would need to be something like $(\uparrow TOPIC) = (\uparrow COMP\, COMP\ \ldots\ OBJ)$, indicating that the topic is linked to the object of *some* complement in the sentence.

The functional uncertainty method allows exactly this last type of equation to be written, using the Kleene-star operator:

(6.3) $(\uparrow TOPIC) = (\uparrow COMP * OBJ)$

Joshi & Vijay-Shanker (1989) shows that a mechanism similar to functional uncertainty can be defined for Feature-Structure-Based Tree-Adjoining Grammars (FTAGs), where the relation between the antecedent and the predicate is again captured in functional terms, but where the mechanism takes advantage of the fact that FTAG 'elementary trees' form a domain for localizing long-distance dependencies.

6.3.3 Valence- and Gap-Filling Algorithms

The previous section discussed ways of *representing* filler-gap relations. This section discusses *algorithms* for combining the filler and the gap in producing a semantic interpretation. In general, gap-filling algorithms fall into one of two classes. In the first class, the *knowledge-based* algorithms, (Fodor 1978; Tanenhaus *et al.* 1985; Ford *et al.* 1982; Hirst 1986; Cardie & Lehnert 1991) the interpreter uses any available knowledge to help decide how to link fillers and gaps. This information can include lexical category, lexical semantics, lexical valence, etc. The second class of algorithms (Clifton & Frazier 1989) assume that the syntactic processor is an autonomous subsystem which assigns fillers to gaps without using any lexical knowledge except perhaps lexical syntactic category information.

In the *lexical expectation model* of Fodor (1978), which she bases on unpublished work by Wanner and others, the processor only proposes a gap for verbs which have expectations for arguments. The subcategorization frame for each verb is ranked, and if the preferred frame for a verb is transitive, the processor hypothesizes a gap following the verb. Thus the model relies on verb subcategorization information.

Boland *et al.* (1989) extend the lexical expectation model to handle multivalent verbs. They suggest that if the verb is multi-valent, the processor first attempts to fill the direct-object role, but if the filler is an implausible direct-object, the processor immediately attempts to fill any other roles the verb has instead, such as indirect-object or infinitive complement. For convenience we will refer to this model as the *multivalent lexical expectation model*. This model is very similar to Sal's *valence integration* algorithm, which is discussed in §6.4.3 below. The major distinction is that since Sal maintains parallel interpretations, integration can proceed on each interpretation simultaneously.

Hirst (1986) proposes a model of gap-filling which evaluates each possible gap location in parallel. In order to consider all the possibilities simultaneously, Hirst's algorithm is not on-line — after the Paragram parser has completed parsing the sentence, it passes all the parses to the Semantic Enquiry Desk. The Semantic Enquiry Desk chooses the filler-gap pairing which is most semantically plausible. Although Hirst's model is not on-line, it is the first implementation which uses plausibility to choose between candidate gap-filler pairs.

Clifton & Frazier (1989) model gap filling with the *Active Filler* process. In their model, whenever a *wh*-element occurs, the processor expects a gap to appear somewhere afterwards, hypothesizing a gap at every syntactically legal position until the gap is filled. Like the Active Filler process, Sal's valence integration algorithm proposes a filler when a construction suggests it, although Sal is somewhat more general in that it proposes a filler not just after *wh-elements*, but whenever the CIG grammar contains a slash-integration declaration, including, for example, topicalization and other such phenomena. The valence integration algorithm also differs from the active filler algorithm in that it searches for semantic gaps, or holes, rather than syntactic gaps, and that it applies semantic constraints on the hole to each filler.

6.4 The Integration Operation

This section describes the details of both *constituent integration* and *valence integration*. Both these kinds of integration are based on a low-level information-combining primitive which is modeled on unification. §6.4.1 defines this primitive operation. §6.4.2 then defines *constituent integration*, and §6.4.3 defines *valence integration*. Finally, §6.4.3 discusses how both constituent integration and valence integration require that a representational distinction be drawn between *constraints* on a filler of some gap or constituent slot, and the actual *filler* of the gap or slot.

6.4.1 The Integrational Primitive

The integration operation combines two informational structures by building a new structure that has all the information from its inputs, augmented with a binding list. As such, it is an extension of the most common information-combining formalisms, *feature unification* and *term unification*. While unification is an adequate operation for combining *syntactic* feature information, models which have attempted to use unification for *semantics* have augmented it with such mechanisms as *lambda-abstraction* and *functional application* (Moore 1989, Pereira & Shieber 1987). The integration operation proposed here also augments the basic unification operation, using unification as a low-level processing primitive. This section describes this primitive operation, based on unification. The following sections will show how this primitive is used in building interpretations.

Like unification, integration combines informational elements by building an output structure which contains all the information from each input structure. Also like unification, it can do this in two ways: by *variable binding*, and by *predicate copying*. In variable binding, a variable in one structure is bound to some part of the other structure. In predicate copying, the new structure is build by explicitly copying predicates from each input structure.

Note that because our semantic structures are expressed in a predicate-calculus-like format, the method of combination is *predicate-copying* rather than *feature-copying*. However, it is always possible to rewrite a predicate-based information-structure as a feature-based one, and so the two methods are basically notational variants. Thus the primitive operation on which the integration operator is built is a version of unification which unifies these predicates. This low-level operation is used as a sub-routine or combinational primitive by the integration mechanism.

The rest of this section will consider three examples of this primitive operation, which are shown in Figure 6.2. In each case the integration operation is represented by the **I** operator, and the product of the operation is shown at the bottom of the figure.

```
    (a Identify $t
       (Unknown $p)
       (Background $x))

    (a Identify $i
I      (Unknown $u)
       (Background $j))
  ----------------------
    (a Identify $t
=      (Unknown $p)
       (Background $x))
    BINDINGS
     (($x $g) ($p $u) ($t $i))

              (a)
```

```
    (a Scale $s
       (On $z))

    (a Scale $x
I      (Domain Red)
       (On $y))
  ----------------------
    (a Scale $s
=      (Domain $g)
       (On $z))
    BINDINGS
     (($z $y) ($s $x))

              (b)
```

```
    (a Identify $t
       (Unknown $p)
       (Background $x))

    (a Means-For $x
I      (Plan $p)
       (Goal $g))
  ----------------------
=        ∅

              (c)
```

Figure 6.2: Three Examples of Integration

Figure 6.2a shows the integration of two assertions which are identical expect for the names of the variables which are bound to them. Each assertion creates an instance of the **Identify** concept, and both have the same slots, which are filled with different variables. The concepts in Figure 6.2a are integrated by first binding together the two concept variables **$t** and **$i**, and then integrating the individual slots of the concepts.

Figure 6.3 shows a raw trace of the integration function which is part of the implementation of the interpreter, in integrating the two structures from Figure 6.2a. The function, called `integrate`, takes two structures and returns a new structure which is the integration of the two input structures. The only result of this simple integration was to bind together a number of variables. Note that the integration function binds together the variables **$t** and **$i**, which are bound to each of the two assertions. In addition, the variables which fill each of the individual slots are also bound together.

Figure 6.2b shows a second example of integration which integrates two structures which each define a *scale* (see Chapter 3). Each of the structures specifies particular information about the scale. When the two structures are integrated the resultant structure specifies a scale which combines the information from the input structures. As before, the concepts in Figure 6.2 are unified by first unifying the two concept variables, and then unifying the individual slots of the concepts. This example will be described further in Figure 6.5 and Figure 6.6.

Figure 6.2c shows an example of a failed integration. Here the two assertions that were passed to the integration function were assertions of different concepts. The first assertion required an instance of the **Identify** concept, while the second assertion required an instance of the **Means-For** concept. Since the integration operation requires that two input assertions have the same concept, this integration failed. Figure 6.4 shows a trace of this failed integration.

```
<cl> (integrate
[(a identify $t
    (unknown $*p)
    (background $x))]

[(a identify $i
    (unknown $*u)
    (background $g))])


(a  identify  $t
    (unknown $p*)
    (background $x*))
BINDINGS:
(($x* $g) ($p* $u*) ($t $i))

<cl>
```

Figure 6.3: A trace of a simple successful integration

6.4.2 Constituent Integration

We turn now to examine the ways in which this primitive integration operation is used in building interpretations. The first process we consider is *constituent integration*. Constituent integration is the name we give the process by which individual constituents of a construction are filled by integration with structures in the access buffer. As mentioned in §4.5.1, constituent integration is very much like a more fine-grained version of the *handle-pruning* mechanisms used by bottom-up parsers (Aho *et al.* 1986). Informally, a *handle* is a substring of the input that matches the right hand side of some rule. Handle-pruning thus consists of replacing a handle in a string with the left-hand side of the relevant rule. In constituent integration, instead of matching the entire right-hand side of a rule with the input, we match a *single constituent* with the input. This is because integration proceeds on a *constituent-by-constituent* basis, instead of the *rule-to-rule* basis which is used in many models of sentence-interpretation (as discussed in §6.2.3).

The *control structure* of constituent integration was described in §4.2, and proceeds by making a copy of each interpretation in the interpretation store, and attempting to integrate it with a copy of each construction in the access buffer. The constituent integration operation itself is thus called on each interpretation-construction pair, and attempts to integrate each construction with the *cursor* of each interpretation. The constituent integration algorithm itself is as follows:

> **Constituent Integration**: Given a construction c which places a set of constraints s on its cursor constituent, and given a proposed constituent g, integrate each assertion in g with each assertion in s, subject to the constraint that s must subsume g.

Figure 6.5 shows an example of constituent integration. The interpretation buffer contains the

```
<cl> (integrate
[(a identify $t
    (unknown $*p)
    (background $x))]

[(a means-for $x
    (plan $*p)
    (goal $g))])

nil
<cl>
```

Figure 6.4: A trace of a failed integration

How-Scale construction. whose cursor selects the proposition **(a Scale $s (On $z)**. The access buffer contains the word *red*.

The concepts in Figure 6.5 are unified by first unifying the two concept variables, and then unifying the individual slots of the concepts. Thus the integration operation acts to match the information in the interpretation cursor with the information in the access buffer. In the case of Figure 6.5, the integration succeeds because both structures contain the same concept, in this case the concept *scale*. In addition, the concept in the interpretation store subsumes the concept in the access buffer.

A raw trace of the output of the integration appears in Figure 6.6. The integration operation performed a number of bindings in integration the two structures. Integrating the two **Scales** required binding together the two variables **$s** and **$x**, while integrating the two **On** clauses required the integration operation to bind the variables **$z** and **$y** together.

The definition of constituent integration above required that the constituent *subsume* any allowable fillers. Thus constituent integration treats constraint information differently than do operations such as *unification*. The integration operation treats constraint information as a set of relations that *must* be present in any gap filler. That is, where unification is *symmetric*, integration is *asymmetric* because it requires that constraints on a constituent *subsume* a proposed filler. In order to fill a gap, a candidate filler must at least include all the semantic relations expressed by the constraints on the constituents. This is true for both *constituent* integration and *constitute* integration. Thus in order for a construction to fill the constituent slot in another construction, i.e., for a construction c to *constituent-integrate* with a constituent t of another construction, t must *subsume* c. See Ingria (1990) for a similar proposal, with a detailed examination of unification and subsumption applied to agreement information in a number of languages.

As §6.2.2 noted, constituent integration can use information about the representation language to decide if a candidate can successfully fill a constituent slot. Chapter 3 noted, for example, that a construction can constrain one of its constituents to be a *weak* construction, such as NOUN or DETERMINER. Recall that each weak construction abstracts over various strong constructions. Since a successful filler must be *subsumed* by the constraints, a constituent which is constrained to be a certain weak construction w can only be filled by constructions which are *subsumed* by

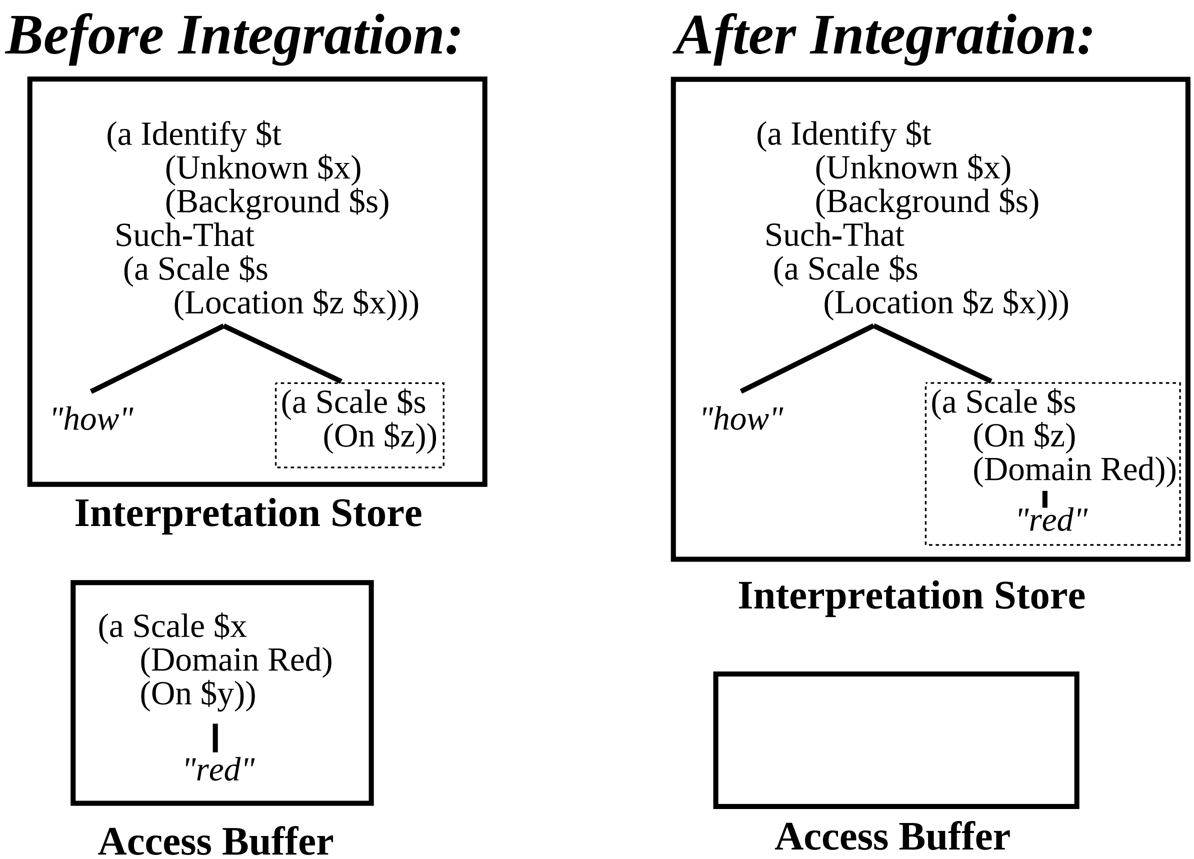


Figure 6.5: Constituent Integration

w. For example if a construction constrains one of its constituents to be a weak construction like DETERMINER, this constituent will integrate successfully with a strong construction like THE, because DETERMINER abstracts over THE. Figure 6.7 below shows a trace of the *constituent-integration* function from the implementation of the interpreter integrating a weak-construction subsumption example. The example begins by setting up a small sample grammar with three constructions. The three constructions are the DETERMINATION construction first introduced in §3.4.2, the lexical construction THE, and the weak construction DETERMINER which abstracts over the THE construction. The trace begins by placing these three constructions in the grammar, and then turns on status reporting and calls the *constituent-integrate* function on the two constructions DETERMINATION and THE. The *constituent-integration* function notes that the DETERMINER construction abstracts over the THE construction, and allows the THE construction to fill the DETERMINER slot in the DETERMINATION construction, binding the variable **$a** to the **Definite-Reference** concept which is the semantics of the THE construction.

```
<cl> (integrate
[(a scale $s
    (on $z))]
[(a scale $x
    (domain $g)
    (on $y)
Such-That
(a red $g))])

(a  scale  $s
    (domain $g*)
    (on $z*)
Such-That
(a  red  $g*))
BINDINGS:
(($z* $y) ($s $x))

<cl>
```

Figure 6.6: A trace of the integration diagrammed in Figure 6.5

6.4.3 Constitute Integration

While *constituent integration* is called by the integration control algorithm to integrate the access buffer into the interpretation store, *constitute* integration is defined explicitly by each grammatical construction. Constitute integration is the means by which the semantics of each of a construction's constituents are combined to build the interpretation for the whole construction. In the simplest case of constitute integration, such as for the HOW-SCALE construction, the semantics of the constitute are build merely by *binding* a variable in the constitute with a variable in one of the constituents. Figure 6.8, repeated from Figure 6.1 above, shows how the semantics of the second constituent are integrated with the semantics of the construction because the variable **$s**, which is bound to the assertion in the second constituent, is also bound to part of the **Identify** assertion in the constitute.

As we discussed above, constitute integration is often more complex than the simple example above because many common linguistic phenomena require elements to be integrated which are not locally instantiated, phenomena such as *valence* (or *subcategorization*), *anaphora*, and other *long-distance dependencies*. Any theory of incremental integration must show how these structures can be built up into interpretations.

In general, such structures occur when some semantic structure must be bound to some variable inside another structure, or (in the vocabulary of the lambda-calculus) one structure must be applied to another. In order to handle these phenomena, the integration operation is augmented with a special operator, called *slash* (at the risk of confusion with the various slash operators of GPSG and Categorial Grammar). Like the slash operator in these two theories, the slash of integration derives from the use of the slash in mathematics to indicate set-subtraction. In the case

```
<cl>
<cl> (begin-grammar)

nil
<cl> (weak Determiner (freq $y)
      [(a Determiner)])

determiner
<cl> (constr determination (freq $p)
  [ (a (Integrate $b $/a))  ]
  ->
  [(a Determiner $a)]
  [(a N $b)])

determination
<cl> (lexicalconstr (The isa Determiner) (freq $f)
 [(a Definite-Reference $ii
     (head $h) )]
->
 [''the''])

the
<cl> (end-grammar)

t
<cl> (setf *debuglevel* 1)

1
<cl> (constituent-integrate 'determination 'the)

Integrate: Construction `determiner' abstracts over construction
`the' of assertion `definite-reference'

Result:
$a*
Bindings
(($a* (!a  definite-reference  $ii
    (head $h))) ($b $/a))

t
<cl>
```

Figure 6.7: Constituent-Integration integrates Weak with Strong Constructions

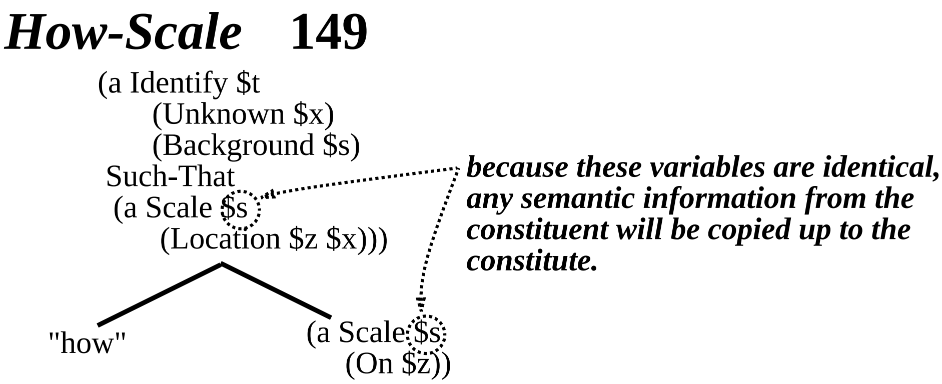


Figure 6.8: The How-Scale Construction Specifies its Integration

of Categorial Grammar and GPSG, X/Y indicated ‘an instance of category X which is missing an instance of category Y’. In the integration theory, a slashed variable indicates that the variable is bound to a structure with a semantic gap (a hole) inside it.

When a slashed structure is integrated with a non-slashed structure, the non-slashed structure is bound to a free variable *inside* the slashed structure. Thus slashing a structure is like *applying* it in categorial or Universal grammar.

The hole-filling integration algorithm can be sketched as follows:

> **Valence Integration Algorithm:** Given a matrix variable m and a filler variable f, examine each hole h_i in m, and when the constraints on a given hole h_n meet the constraints on the filler f, integrate h_n with f. If there is no such hole h_n, but some part of the matrix m is still incomplete, wait and try again.

This algorithm allows the the grammar to specify ways in which information-combination can occur over a distance. In later sections we will discuss a number of grammatical constructions which require such distant instantiation. For the rest of this section, however, we will consider simpler structures involving verbal valence (i.e., information in the verbal lexical entry on subcategorization, thematic roles, and semantic arguments).

As we saw in §3.8, the lexical entry for valence-bearing words like verbs includes information on the possible arguments the verb can take, including their number, and any semantic and syntactic constraints on them. Each of these arguments is represented as a hole in the lexical structure. Just as in unification or lambda-calculus-based approaches, the verb thus acts as a function which is applied to its arguments, with the extension that the semantic predicates which define the hole act as constraints on any fillers.

If the semantic gaps are represented in the verbal lexical entry, and the fillers are noun phrase or prepositional phrases, the grammar requires a third construction which specifies how these two are integrated together. This is the verb-phrase construction. There are many verb-phrase constructions in our grammar — we begin with the MONO-TRANSITIVE-VERB-PHRASE CONSTRUCTION. This verb-phrase construction accounts for transitive verb-phrases with a single complement. Figure 6.9 shows a representation of the construction.

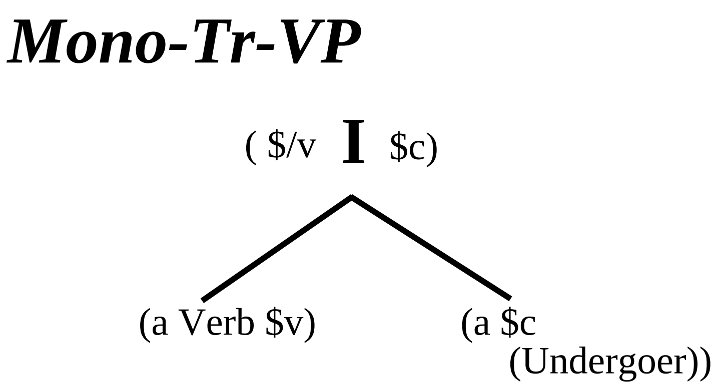


Figure 6.9: The Mono-Transitive-Verb-Phrase Construction

Note that the symbol **I** is used to indicate the integration operation. Thus the MONO-TRANSITIVE-VERB-PHRASE construction builds its semantics by integrating its two constituents, **$v** and **$c**. The **Verb** constituent in the integration has been marked with a slash (**$/v**). This indicates that the verb will serve as the matrix for the complement. The second constituent of the construction, labeled **$c**, has been constrained to fill the **Undergoer** role (Foley & van Valin 1984), which abstracts over those thematic roles which generally act as grammatical objects. This will constrain which valence role of the verb will be filled by the integration. More details on valence semantics are in §3.8.

Let's trace the operation of the integration function on the MONO-TRANSITIVE-VERB-PHRASE construction just defined. Consider the sentence in (6.4):

(6.4) *Casey hit the ball.*

The DECLARATIVE-CLAUSE construction was used to link the subject **Casey** to the verb *hit*. Thus just before the noun-phrase *the ball* is interpreted, the state of the construction appears as in Figure 6.10.

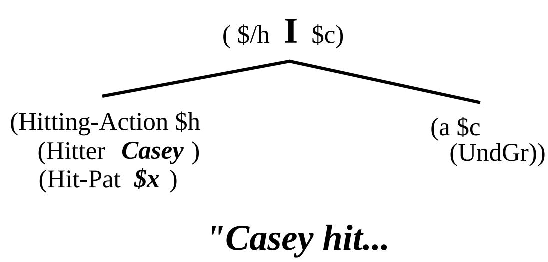


Figure 6.10: Interpreting "Casey hit. . . "

When the phrase *the ball* is first constituent-integrated, but before the valence integration is done, the construction appears as in Figure 6.11.

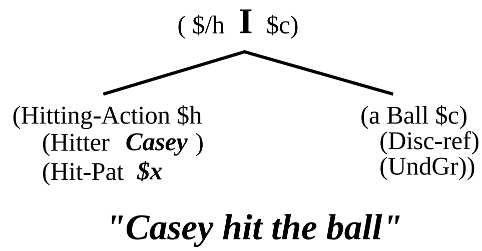


Figure 6.11: Interpreting "Casey hit the ball (1)"

In order to build the semantics for the MONO-TRANSITIVE-VERB-PHRASE construction, the integration operation must find a gap in the structure bound to the variable **$h**, i.e., the semantic structure of the **Hitting-Action** concept. There is only one unbound variable in this structure — the variable **$x**, the filler of the **Hit-Patient** slot. In order for integration to succeed, however, the constraints on the variable **$x** must match the constraints on the filler **$c**. This filler, the **Ball** object, is constrained to be an **Undergoer** by the verb-phrase construction. The **Hit-Patient** concept is indeed defined to be an acceptable **Undergoer**, (see §3.8), and hence the integration proceeds as shown in Figure 6.12.

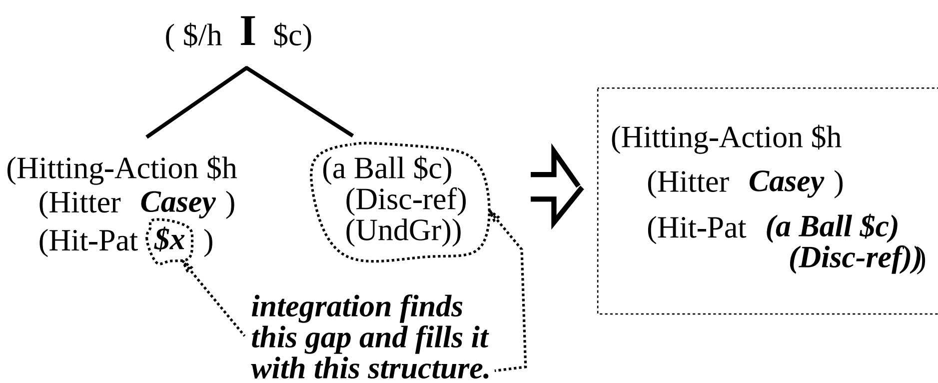


Figure 6.12: Interpreting "Casey hit the ball (2)"

The valence integration algorithm is thus *knowledge-intensive*, in that a variable may be constrained by any type of linguistic knowledge — grammatical category, semantics, control information. This aspect of the algorithm is compatible with a broad class of psycholinguistic results. Mitchell & Holmes (1985), for example, shows that integration is able to use lexical information like syntactic category and subcategory. Shapiro *et al.* (1987) shows that integration can also use lexical semantic information. Boland *et al.* (1990) and Tanenhaus *et al.* (1989) show that when a verb specifies control information for its verbal predicates, this information is also used by the integration mechanism, and indeed is available immediately.

Marking Variables

Expectations plays a strong role in the design of this interpreter. For example, §5.2 showed that two kinds of expectations can be used to help access constructions: *constituent-based* expectations and *valence-based* expectations. Constituent-based expectations arise from the constraints that a construction places on its constituents, while valence-based expectations arise from the constraints that a lexical construction places on its *valence arguments*. [3]

The integration theory allows each of these kinds of expectations to constrain the integration process. Thus constituent-based expectations constrain *constituent integration*, while valence-based expectations constrain *constitute integration*. But in making these integrations, the interpreter must be able to distinguish between these kinds of *constraints* on gaps, and information which is actually present in the gap *filler*. Symmetric, monotonic operations such as unification do not allow these two kinds of information to be distinguished — both are represented uniformly as attribute-value pairs. Thus in a grammar which uses simple unification, for example, there is no way to know whether a particular verbal subcategorization or valence argument has been filled.

Other models handle this problem by extending the unification operation to allow a particular atomic value with a special interpretation. The **ANY** value of Functional Unification Grammar, discussed in Shieber (1986), acts like a variable in that it will unify with any variable. However, an **ANY** value which does *not* unify with another variable or feature structure marks an unfilled argument of a verb.

Rather than propose a special atomic value, the integration theory described in this dissertation makes a representational distinction between gap-constraints and gap-fillers. We do this by *marking* variables which have been filled. *Unmarked* variables indicate *constraining* information, while *marked* variables indicate the *filler* of a gap.

The integration operation takes advantage of this representational difference in combining information. For example, when the valence integration algorithm is searching for a semantic gap, it only considers unfilled gaps, i.e., those gaps whose variables are *unmarked*. Consider briefly an example of valence integration that was presented in §4.9, in which the valence integration algorithm is looking for a semantic gap in the MEANS-HOW construction (this construction is defined in §3.8.2).

Figure 6.13 might be paraphrased in English as "a question about the means **$p** for achieving some goal **$g**". As §3.8.2 discusses, this lexical item has a single valence argument, the **Goal $g**. In order for the integration algorithm to realize that this structure only has a single valence argument, each variable which is *not* a valence argument must be *marked*. The following table shows the state of each of the variables in Figure 6.13:

[3] Although CIG is currently only embedded in a model of *interpretation*, it might be suggested that in a model of *production*, an asymmetry in the part-whole structure of the construction might be reversed. For interpretation, the information in the constituent slots of the construction definition is interpreted as *constraints* on candidate constituents, while the information in the *constitute* element is interpreted as *instructions* for creating a whole semantic structure. For production, we might expect that the *constitute* imposes constraints on what concepts may be expressed by the construction, while the constituents give instructions for how to build structure.

```
        (a Identify $t
            (Unknown $*p)
            (Background $x)
Such-That
        (a Means-For $x
            (Means $p)
            (Goal $g)))
```

Figure 6.13: The Semantics of the MEANS-HOW construction

Variable Bindings	
$t	marked as an instance of the Identify concept
$x	marked as an instance of the Means-For concept
$p	marked as an open variable
$g	unmarked

The chart shows how all the other variables in Figure 6.13 are marked. The variables *$t*, and *$x* are marked because they are bound to instances of concepts by the assertion operator **a**. As §3.8 discusses, the operator **a** creates an individual, and thus the variable it fills is not an open valence argument. The variable *$p* is previously marked by the **Wh-Non-Subject-Question** construction as being obligatorily open inside this construction. This marking is done in the grammar, because the questioned element of a question, (in this case the **Unknown** element of an **Identify**) is not allowed to be filled by the question. The questioned element acts as an *open variable* in the discourse, and thus cannot be filled by the question itself.

Thus the only unfilled variable in this structure is the unmarked variable *$g* which fills the **Goal** relation. The traces of the interpreter which are included in this dissertation generally distinguish marked variables by marking them with an asterisk (i.e., **$*x**).

§6.5 shows how the valence-integration algorithm accounts for a number of grammatical phenomena where information-combination is not local.

6.5 Integrating Slashed Elements

The various constructions which are subsumed under the modern term *long-distance dependencies* (wh-movement, topicalization, right-node raising, heavy-np shift, etc.) have caused most linguistic or computational theories to propose special mechanisms to handle them. The *traces* of GB (and HPSG), the *slash-categories* of GPSG and HSPG, the *functional uncertainty* of LFG, the *hold mechanism* of ATNs and the *adjoining* operation of TAGs were all proposed to enable the information from the distant element to be combined with the rest of the information of the clause.

In our model of linguistic knowledge and interpretation, there is no need for a distinct mechanism to interpret long-distance dependencies. The same integration operation which combines elements in a simple construction and combines valence-bearing elements with their arguments also combines distant elements.

This section will show a number of examples of the use of slashed variables in integration, and show how the valence-integration algorithm can handle the integration of filler-gap dependencies in a general way that is consistent with a number of psycholinguistic results.

What is novel about using the integration mechanism to combine long-distance elements is that the combination is done semantically; the grammar does not use syntactic traces, empty categories, or coindexing as place-holders for semantic integration. Fronted elements are integrated directly with the clauses with which they are semantically related. In contrast, all of the theories mentioned above require some sort of mediating syntactic or functional coindexing, or even phonologically null elements in the syntactic structure of a sentence.

Figure 6.14 compares our treatment of long-distance dependencies with the traditional empty-category model for the sentence *What did George take from the fridge?*. The empty-category models postulate a wh-trace directly after the verb *take* which is coindexed with the wh-element *what*. In our model, the wh-element *what* is integrated directly into the semantic structure of *take*.

Syntactic Gap Coindexing:

What $_i$ did George take *wh-trace* $_i$ from the fridge?

Semantic Gap Integration:

Identify
Unknown *i*

Take
Taker: *George*
Taken: *i*

What did George take from the fridge?

Figure 6.14: Integrating

One clear advantage of dispensing with complex syntactic long-distance dependency mechanisms is parsimony. Because the interpreter must produce an incremental semantic interpretation of the sentence anyway, there is no cost in using the semantic gap-filling mechanism instead of a syntactic one. This makes our integration method more parsimonious than syntactic ones in two ways. First, using integration with semantic gaps allows long-distance dependencies to be treated with the same mechanism that is used to do all other semantic integrations. Second, because gaps are semantic rather than syntactic, there is no need for the grammar to have special syntactic mechanisms to handle long-distance dependencies (such as those mentioned above).

The next section §6.5.1 presents an example of the representation and integration of the *non-subject gaps* by the WH-NON-SUBJECT-QUESTION construction. The following sections, §6.5.2–§6.5, present examples which show how the model is consistent with a number of psycholinguistic results.

6.5.1 Slash Integration — An Example

This section discusses the representation and integration of the two constituents of the WH-NON-SUBJECT-QUESTION construction (Jurafsky (1990) and (1991)), which was introduced in §3.6. This construction accounts for sentences which begin with certain wh-clauses, where these clauses do not function as the subject of the sentence. Examples include:

(6.5) a. **How** can I create disk space?

b. **What** did she write?

c. **Which book** did he buy?

The construction has four constituents. The first, indicated in bold type in the examples above, is a wh-element, specified as an instance of the **Identify** concept (see §3.6). The second is an auxiliary verb, and participates together with the third constituent in the SUBJECT-PREDICATE construction, while the second and fourth constituents are constrained to occur in an instance of the VERB-PHRASE construction, The representation for the construction appears in Figure 6.15 below.

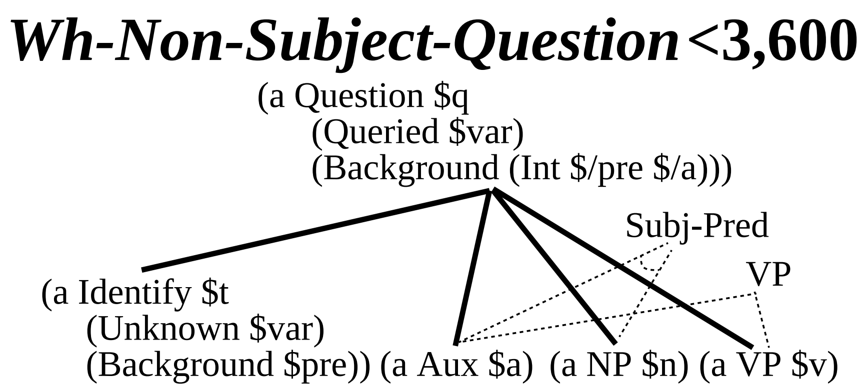


Figure 6.15: The Wh-Non-Subject-Question Construction

Note in Figure 6.15 that the background knowledge for the question is formed by integrating the variables $pre and $a. These contain the information from the two constituents. Note also that each of these variables is slashed. The fact that both variables are slashed indicates that the semantic gap could be in the structures bound to either of these variables. The gap could be inside the semantics bound to the AUX constituent, or inside the **Identify** structure.

For example, in the sentence "*What did she write?*" the gap is located in the fourth constituent, the VERB-PHRASE, because the verb "write" has an unfilled semantic slot for the object written. The integration algorithm will bind the semantics of "what" to the unfilled "written-object" slot of the verb "write".

Consider now the interpretation of the sentence "*How can I create disk space?*". This sentence includes an instance of the MEANS-HOW construction defined in §3.8.2. The MEANS-HOW construction is concerned with the *means* of some action, asking for a specification of the means or plan by which some goal is accomplished.

***Means-How* <675**

```
(a Identify $t
     (Unknown $p)
     (Background $x)
 Such-That
  (a Means-For $x
       (Means $p)
       (Goal $g)))
          |
        "how"
```

Figure 6.16: The Means-How Construction

In (6.5a) the gap is in the *first* constituent, in the MEANS-HOW construction. As §3.8.2 discussed, the gap in this construction is the **Goal $g**.

(6.6) shows the semantics of the second constituent, the CAN construction:

(6.6)
```
(a Ability-State $x
   (Actor $a)
   (Action $b))
```

In order to build the correct interpretation of the sentence, the integration algorithm realizes that the **Goal $g** in Figure 6.16 is a semantic gap which can be filled by the Ability-State $x in 6.6, and it binds the Ability-State to the variable $g. The final result of the integration of the sentence is presented in Figure 6.17.

As we discussed above, the gap in the sentence "*How can I create disk space*" is in the word "how" rather than in the **Subject-Second-Clause**. Other linguistic analyses require wh-phrases to fill a syntactic gap in the matrix clause, which requires them to include traces or empty categories corresponding to each possible syntactic modifier position in the **Subject-Second-Clause**. By placing the gap inside the semantics of "how", we eliminate these numerous empty categories.

Figure 6.18 compares the CIG model, in which the semantic gap is located in the MEANS-HOW construction, with the empty-category model.

6.5.2 Semantic Gap-Filling

This section discusses evidence that gaps are filled by their antecedents directly at the verb (or other valence-bearing element) rather than mediated by a syntactic trace or empty category. These include a number of studies showing the 'filled-gap effect'. This occurs when the processor has found an antecedent and expects it to fill a valence-gap in the verb (or in traditional models to fill a trace after the verb), but this argument position/trace position is already filled. Consider examples (6.7) — (6.8), modified from Fodor (1989):

(6.7) Who$_i$ could the little child have forced$_i$ to sing those stupid French songs for Cheryl last Christmas?

```
(a Question $q
  (Queried $p)
  (Background
    (a Means-For
      (Means $p)
      (Goal
        (a Ability-State  $x
           (Actor (a Speech-Speaker))
           (Action
               (a Creation-Action
                   (Actor (a Speech-Speaker))
                   (Theme (a Disk-Freespace)]
''A question about the means for achieving the
goal of being able to create some disk space.''
```

Figure 6.17: The Semantics of 'How can I create disk space?'

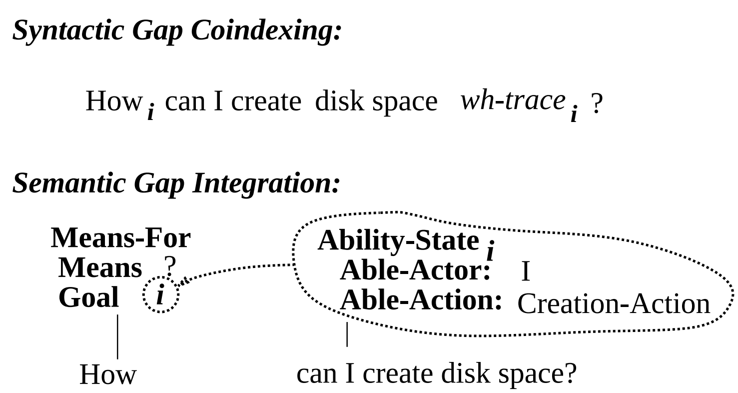


Figure 6.18: The Semantic Gap is Not in the Verb

(6.8) Who$_i$ could the little child have forced us to sing those stupid French songs for$_i$ last Christmas?

Note that in (6.7), there is no direct object after *forced*, while in (6.8, there is an explicit direct object *us*. Response times from Crain & Fodor (1985) showed that (6.8) is more difficult to process than (6.7). This difficulty is manifested exactly at the position of the "filled-gap", i.e., at the word *us* in (6.8). In examples like these, the processor seems to be integrating the antecedent directly into the verb, and is therefore 'surprised' when attempting to fill the already-filled gap with *us* in (6.8). A number of experiments have demonstrated this "filled-gap effect", including Crain & Fodor (1985), Stowe (1986), and Tanenhaus *et al.* (1989).

Another group of results using the *embedded anomaly* technique, in which the antecedent is a semantically implausible filler for the verb, show effects exactly at the verb. These include

Garnsey *et al.* (1989) and Tanenhaus *et al.* (1989).

The filled-gap and embedded-anomaly results support our model in which the antecedent is directly integrated with the verbal valence frame. But these results are also compatible with an empty-category model, in which there is a trace located directly after the verb *forced* in (6.7).

The reason the results do not distinguish the two models is that the antecedents which were considered in these experiments all corresponded to direct-objects, and hence the traces were located directly after the verb. In order to distinguish the empty-category model from the direct-integration model, it is necessary to consider cases where the verb and the syntactic trace are separated by intervening material. Anomaly effects which happen directly or soon after the verb would then provide evidence for the integration theory proposed here. Anomaly effects which do not occur until later, that is until the trace position, would be evidence for the empty-category hypothesis.

Boland & Tanenhaus (1991) studied exactly such a case, in which subjects were asked to process sentences with antecedents for indirect objects. They used examples like (6.9)–(6.10), in which both antecedents are sensible indirect objects of *distribute*, but only *pupils* is a sensible indirect object of *distribute science exams*. That is, the direct object *the science exams* is compatible with (6.9) but not (6.10). Thus the direct-integration model predicts that the anomaly will show up when *science* is processed. The empty category model will predict that the anomaly will not show up until after the preposition *to*, when the trace occurs. Tanenhaus and Boland found that in fact the anomaly showed up in reading time and in semantic acceptability at the presentation of the word *science*. This shows that gap-filling must have taken place by that time, and hence must take place directly at the verb, and not mediated by a syntactic trace.

(6.9) Which uneasy pupils did Harriet distribute the science exams to in class?

(6.10) Which car salesmen did Harriet distribute the science exams to in class?

Pickering & Barry (1991) also present a number of examples where the empty-category model would locate a gap extraordinarily far from the verb, arguing from the general on-line nature of interpretation that gap-filling must proceed by a direct association between antecedent and filler.

6.5.3 WH-Questions and WH-Subordinate-Clauses

In general, as the last section noted, finding empirical evidence that distinguishes between our integration model and the traditional empty-category model is difficult. This is because the empty-categories proposed by these models are usually placed directly after the valence-bearing element — thus both models predict that 'gap-filling' or integration will occur after the onset of the predicate and before the onset of the next word.

Distinguishing the models requires considering cases where the hypothetical empty-category is not located directly after its subcategorizer. The SUBJECT-WH-QUESTION is such a construction, because the classical model would place an empty category *before* rather than *after* the verb. In (6.11a)–(6.11c) below, theories with empty categories would insert them after the bold-face wh-elements that begin each of the examples:

(6.11) a. **Who** invented the airplane?

b. **What** is for dinner?

c. **Which book** explains the meaning of life?

Figure 6.19 shows a representation of the Wh-Subject-Question construction.

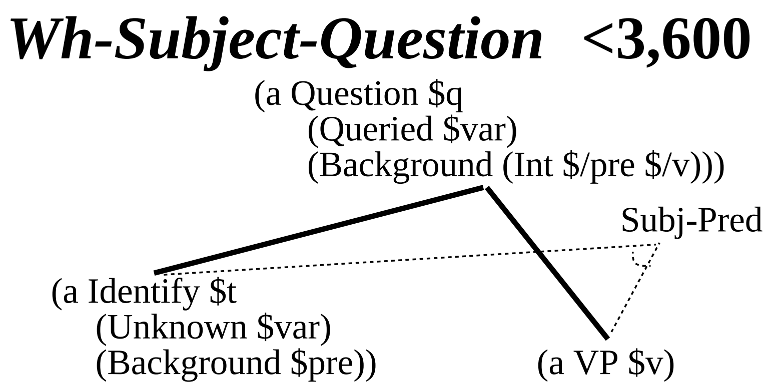


Figure 6.19: The Wh-Subject-Question Construction

Note that the Wh-Subject-Question construction has two constituents. The first one, like the Wh-Non-Subject-Question, is constrained to be an instance of the **Identify** concept. Recall from §3.6 that the **Identify** concept characterizes the *wh*- constructions — it instantiates a frame in which the identity of some element is in question, and where some background information is provided to help identify the element. The second constituent of the Wh-Subject-Question construction is constrained to be a verb-phrase. Note that the first constituent, the **Identify** element, is then constrained to be the subject of this verb phrase.

Three important psycholinguistic results related to the Wh-Subject-Question and the Wh-Subject-Subordinate-Clause construction were presented in Stowe (1986). Stowe's first results concerned the ease or difficulty of processing **Wh-Subject-Subordinate-Clauses** versus **Wh-Non-Subject-Subordinate-Clauses**. In particular, she was attempt to see if the Crain & Fodor (1985) "filled-gap" results concerning traces in object position extended to traces in subject position. Recall from §6.5.2 that Crain and Fodor showed that the processor used the wh-antecedent to predict an upcoming post-verbal gap, and experienced difficulties if the gap was already filled.

Stowe showed that this was *not* the case for subject gaps, that in fact people did not experience any difficulty when encountering an explicit subject, even when they were expecting a subject gap. She suggests that this may imply that the algorithm for interpreting wh-gaps in subject position is very different from the one for interpreting gaps at object position.

Stowe's results supports the direct-integration view of gap-filling presented here, and seem to provide counter-evidence to the empty-category view. This is because the empty-category model would need to assume different processing for subject as against object gaps, which would be rather difficult as empty-category theories do not distinguish subject and object gaps.

In CIG, on the other hand, subject gaps and object gaps are quite distinct; they belong to different constructions. Thus it is quite possible that their processing will be different. Consider examples (6.12a)–(6.12c) from Stowe (1986):

(6.12) a. My brother wanted to know who — will bring us home to Mom at Christmas.

b. My brother wanted to know who Ruth will bring — home to Mom at Christmas.

c. My brother wanted to know who Ruth will bring us home to — at Christmas.

Stowe showed first that readers had no difficult reading 'filled' subject gaps, like the noun-phrase *Ruth* in (6.12b) or (6.12c). Her second finding was that readers *did* have trouble with 'filled' object gaps, like the noun-phrase *us* in (6.12c). Finally, readers had no trouble with 'filled' object gaps like *us* in (6.12a) if they occurred after a *wh*-element had *already* filled a gap.

Each of these results would be expected if the interpreter processes *wh*-elements as CIG and the integration theory predict. Consider the state of the interpreter directly before processing the word *Ruth* in the fragment (6.13) which begins each sentence in (6.12) above.

(6.13) My brother wanted to know who . . .

The interpretation store will contain two constructions — the WH-NON-SUBJECT-SUBORDINATE-CLAUSE construction and the WH-SUBJECT-SUBORDINATE-CLAUSE construction, since the input up to that point does not distinguish between the two. The next word, *Ruth* or *will*, distinguishes between the interpretations — *Ruth* (as in (6.12b) or (6.12c) is consistent with the WH-NON-SUBJECT-SUBORDINATE-CLAUSE construction, while *will* (as in (6.12a)) is consistent with the WH-SUBJECT-SUBORDINATE-CLAUSE construction. Neither word should cause any problem or re-analysis for the interpreter; either one will simply cause one interpretation to be selected. This is consistent with Stowe's first result that subject gaps did not cause any processing difficulty. Figure 6.20 shows this state of the interpreter.

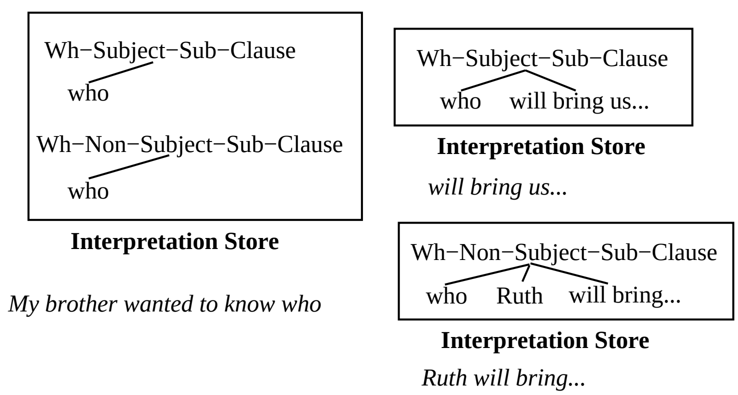


Figure 6.20: Interpreting Subject Versus Object Gaps (1)

But now consider the processing of object gaps. Recall that Stowe second finding was that (6.12c) caused the 'filled-gap' effect at the word *us*, presumably because the object gap was already filled by the *wh*-element *how*. The integration algorithm can model this result as well. An object gap, such as occurs in (6.12b) above, can only occur once the WH-NON-SUBJECT-SUBORDINATE-CLAUSE has already been selected, as in the bottom interpretation in Figure 6.20.

When the interpreter gets to the word *bring* the integration operation will integrate the *wh*-element *who* into the semantics of *bring*. This is because the WH-NON-SUBJECT-SUBORDINATE-CLAUSE construction specifies that the *wh*-element is integrated into the VP constituent. Now when the interpreter sees the word *us*, it must re-analyze the interpretation, and undo the integration of *who* into *bring*, because *us* is interpreted as the Goal of *bring*. Figure 6.21 shows the state of the interpreter just before and after seeing the word *us*.

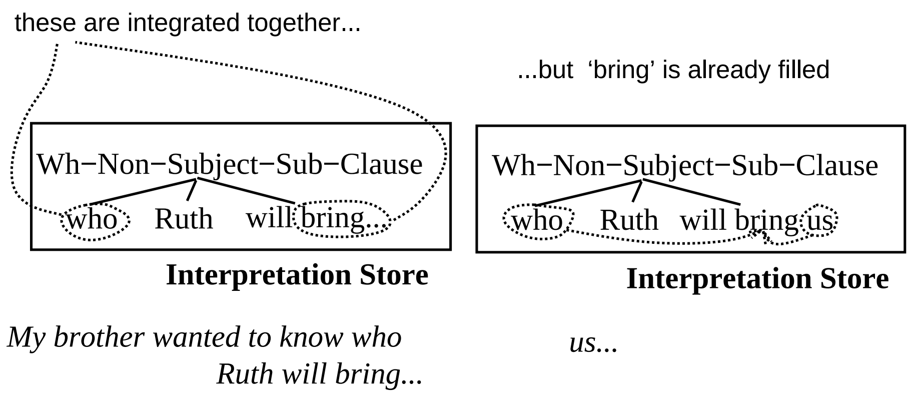


Figure 6.21: Interpreting Subject Versus Object Gaps (2)

Stowe's third result was that readers had no trouble with 'filled' object gaps like *us* in (6.12a) if they occurred after a *wh*-element had *already* filled a gap. That is, the processor stops looking for a gap once it finds one. This is in fact how the valence integration algorithm works; once a gap is filled, the operation does not continue attempting to find a binding.

Chapter 7

The Selection Theory

In defining a theory to guide an interpreter like Sal in choosing among possible interpretations, we might do well to recall that familiar philosophical mammal, Buridan's ass. Remember that Buridan's ass was placed exactly in between two equal portions of hay. With no metric allowing him to choose between the hay on the left or the hay on the right, the ass is in danger of starving, because he cannot decide toward which portion to move. The ass must decide *how* and *when* to choose a bale of hay. These same questions in our model of sentence interpretation, as follows:

How do we choose among interpretations?
When do we choose among interpretations?

7.1 A Sketch of a Selection Algorithm

Let us begin with an answer to the first question. Like the donkey, we would like to choose the larger pile of hay — i.e., the interpretation which is highest on some metric. The metric we choose is coherence with contextual expectations.

In describing the access and integration mechanisms we have focused on the importance of *expectations* in suggesting constructions and combining them into interpretations. These expectations are as important in selecting among interpretations as they are in creating them. Indeed, the selection theory which I will describe here is based simply on assigning each candidate interpretation a confidence measure based on its coherence with various kinds of expectations. The theory assigns preferences to interpretations by the **Selection Choice Principle**:

> **Selection Choice Principle:** Prefer the interpretation whose *most recently integrated element* was the most *coherent* with the interpretation and its lexical, syntactic, semantic, and probabilistic expectations.

The Selection Choice Principle refers to a number of kinds of expectations. The term "expectation" has been used most frequently to mean the sort of slot-filling processing that is associated with the scripts of Schank & Abelson, the frames of Minsky, the schemas of Bartlett, even the *noema* of Husserl. The term is used for similar purposes in the selection theory. Selection theory expectations include *constituent expectations*, which are expectations which a grammatical construction has for particular constituents, *valence expectations*, which are expectations that particular lexical items have for their arguments, as well as *frequency expectations*, based on the idea mentioned in Chapter 5 that more frequent constructions are more expected than less frequent constructions. As Chapter 3 discussed, each construction is annotated with a relative frequency, drawn from its occurence frequency in the Brown Corpus.

Of course the selection choice principle will not be sufficient to solve every case of disambiguation — clearly disambiguation is a process that must refer to every level of linguistic knowledge, including pragmatic and textual knowledge which is not considered in this thesis, as well as non-linguistic world knowledge. As Hirst (1986:111) noted, it is impossible to disambiguate sentences like (7.1a,b) without non-linguistic knowledge about "the relative aesthetics of factories and flora":

(7.1) a. The view from the window would be improved by the addition of a <u>plant</u> out there.

b. The view from the window would be destroyed by the addition of a <u>plant</u> out there.

But the use of grammatical expectations derived from lexical semantics, valence constraints, syntactic constituency constraints, and constructional frequencies, is a necessary part of any disambiguation model. Indeed, as Norvig (1988) showed, a selection theory which simply chose the most plausible interpretation would fail to meet the constraints of cognitive validity. (7.2) lists a number of well-known examples in which people choose implausible interpretations in order to fill local expectations, or in which local preferences cause correct interpretations to be discarded in favor of incorrect ones:

(7.2) a. The landlord painted all the walls with cracks.

b. The horse raced past the barn fell *(from Bever (1970))*

c. The prime number few. *(from Milne (1982))*

d. Ross baked the cake in the freezer. *(from Hirst (1986))*

In each of these cases, the reader initially arrives at an interpretation which is semantically anomalous. This misreading is due to local grammatical expectations, which override, at least temporarily, the more global semantic well-formedness of the sentence. Sal's selection mechanism emphasizes examples like those in (7.2) which exhibit this *local coherence* phenomenon.

In addition, of course, even if we had good formal theories of the representation of aesthetic knowledge, some principle such as the Selection Choice Principle would be needed to show how the expectations derived from this knowledge can be used to inform future selection choices.

The interpreter solves the second problem (*when* to choose) by assuming that because the interpreter's *working store* is limited like human short-term memory, interpretations are pruned whenever they become significantly less-favored than the most preferred interpretation. §4.7 showed that forcing selection to be *on-line* in this manner also solves some long-standing efficiency problems in parsing. The timing constraint is stated in the Selection Timing Principle:

> **Selection Timing Principle:** Prune interpretations whenever the difference between their ranking and the ranking of the most-favored interpretation is greater than the selection threshold σ.

The Selection Timing Principle requires that an interpretation is pruned whenever there exists a much better interpretation. When all of the alternative interpretations have been pruned, the most-favored interpretation will be selected. Thus the interpretation store may temporarily contain a number of interpretations, but these will be resolved to a single interpretation quite soon. The point at which one interpretation is left in the interpretation store is called the *selection point*. Like the *access point* of Chapter 5, the selection point is context dependent. That is, the exact time when selection takes place will depend on the nature of the candidate interpretations and the context. Just as the *access threshold* α was fixed but the *access point* was variable, the *selection threshold* σ is fixed, while the *selection point* will vary with the context and the construction. The *selection point* resembles the *recognition point* which is used to define the point of final lexical selection in the Cohort model (Marslen-Wilson 1987).

The remainder of this chapter will explore the Selection Choice Principle and the Selection Timing principle in further detail. §7.2 summarizes the criteria which are used for ranking interpretations. §7.3 explains the Selection Timing Principle, and shows how it can be implemented by extending the model of Gibson (1991). §7.4 surveys previous selection models, and §7.5 discusses the use of locality in selection. Finally, §7.6 shows that a number of well-known cases of ambiguity can be correctly disambiguated by the Selection Choice Principle.

7.2 The Selection Choice Principle

The Selection Choice Principle simply says to choose the interpretation which is most *coherent* with grammatical expectations. The workings of this principle, then, depend on the ranking algorithm which is use to measure coherence with expectations. This section proposes a specific ranking algorithm based on the integration operation — interpretations are ranked according to how well their *latest integration* was coherent with grammatical expectations and the rest of the interpretation.

The ranking algorithm can be summarized very simply by considering three possible disambiguation situations:

- Given a choice between interpretation whose last integration filled an expectation, and one whose last integration did not, the selection algorithm will prefer the former.
- Given a choice between two interpretations *both* of whose last integrations fill expectations, the selection algorithm will prefer the one which filled the *stronger* expectation.
- Given a choice between two interpretations *neither* of which filled an expectation, the selection algorithm will prefer the interpretation which has integrated the elements in the access buffer to one that has not.

This description makes use of a *strength* ranking for interpretation coherence. Coherence is defined according to the following ranking:

The Coherence Ranking: (in order of preference)

I Integrations which fill a *very strong* expectation such as one for an exact construction, or for a construction which is extremely *frequent*.

II Integrations which fill a *strong* expectation such as a *valence expectation* or a *constituent expectation*.

III Integrations which fill a *weak expectation*, such as for an optional adjunct or include feature matching rather than feature imposing.

IV Integrations which fill no expectations, but which are nonetheless successfully integrated into the interpretation.

V Integrations which are *local*, i.e., which integrate the elements which are the closest together.

VI Integrations which fill no expectations, and are not integrated into the interpretation.

It is important to note once again that coherence with expectations is not being offered as a complete solution to the problem of selecting among ambiguous input. Besides the need for a notion of 'real-world or textual-world plausibility', this algorithm clearly needs to be more fine-grained, and needs to account for expectations based on such knowledge as previous discourse referents (such as is proposed by Crain & Steedman (1985) and used by Hirst (1986)). But that being said, this simple model does account for a great deal of selection preferences.

The rest of this section will continue the exposition of the expectation ranking and the Selection Choice Principle by examining specific cases where the interpreter uses the ranking to select among interpretations at different levels. Examples will be drawn from the more detailed studies in §7.6.

Strong Expectations

The most obvious corollary to the Selection Choice Principle is the commonly noted preference for verbal arguments over verbal adjuncts. Every major model of selection includes some way to account for this preference. The Selection Choice Principle accounts for this in a more general fashion, however. The interpreter will prefer an interpretation which fills *any* expectation to one that fills no expectation. This includes verbal valence expectations, thus preferring verbal arguments over verbal adjuncts, but also includes nominal valence expectations, and constituent expectations.

An example of a constituent expectation occurs with the SUBJECT-EXTRAPOSITION construction (the details of Sal's processing of this construction are presented in §7.6.4). Crain & Steedman (1985) noted that when processing extraposed clauses such as (7.3), people prefer to analyze the clause *John wanted to visit the lab* as a complement clause rather than as a relative clause modifying *the child*.

(7.3) It frightened the child that John wanted to visit the lab.

We can see how this preference would be predicted by the Selection Choice Principle by considering the two candidate interpretations of the sentence just after processing the word *"child"*. There are two candidate interpretations at this point, one involving the DECLARATIVE-CLAUSE construction, and the other the SUBJECT-EXTRAPOSITION construction. In the DECLARATIVE-CLAUSE interpretation the word *it* acts as a normal pronoun, and there are no unfilled verbal or constructional expectations. Although the word *"that"* could begin a post-nominal relative clause, there is no expectation for it. The SUBJECT-EXTRAPOSITION interpretation, however, does have one unfilled constituent slot — the slot for a SUBORDINATE-PROPOSITION, which begins with the word *"that"*. (The SUBJECT-EXTRAPOSITION construction has three constituents — the subject *it*, a VP, and a SUBORDINATE-PROPOSITION.)

Because the word *"that"* fills an expectation in the SUBJECT-EXTRAPOSITION construction but not in the DECLARATIVE-CLAUSE construction, the SUBJECT-EXTRAPOSITION construction is preferred. Thus when choosing between an interpretation which fills an expectation and one which does not, the expectation is preferred.

Very Strong Expectations — Specificity

In the case of (7.3), only one of the possible interpretations was expected. If both interpretations are produced by filling an expected argument, the selection algorithm will prefer the *strongest* expectation according to the ranking criteria above. The most highly ranked expectations are called *very strong* expectations. An expectation is *very strong* when compared with another if it constrains its filler more specifically than the other expectation, or if its filler is much more frequent than the filler of the other expectation. Thus given a choice between two expectations, if one is more specific to the constituent just integrated, it is selected. The idea of choosing a more specific rule when two rules apply is often referred to as *Panini's Principle*, and was proposed by Wilensky & Arens (1980) and Wilensky (1983) for choosing among interpretations, and by Hobbs & Bear (1990) for choosing among attachments.

Consider, for example, the ambiguous phrase *grappling hooks* in (7.4) from Milne (1982) in which the word *hooks* can function as a noun (as in (7.4a)) or a verb (as in (7.4b)):

(7.4) a. The grappling hooks were lying on deck.

b. #The grappling hooks on to the enemy ship.

The use of *hooks* as a noun, as in (7.4a), is much preferred. Milne (1982) found that sentences like (7.4b) cause processing difficulty. The preference for (7.4a) falls out of the Selection Choice

Principle because *grappling hooks* is a collocation — that is, there is a specific construction GRAPPLING-HOOKS which has two constituents, the first *"grappling"*, and the second *"hooks"*. Because the construction is a lexical one, it has a very strong (lexical) expectation for the word *"hooks"*. Thus when *hooks* appears, it meets this strong expectation. In (7.4b), on the other hand, the SUBJECT-PREDICATE construction only gives rise to an expectation for a VERB — i.e., for any verb. This expectation is not a very specific one; there are a great number of verbs, and therefore by the Coherence Ranking, it is not as strong an expectation as that from GRAPPLING-HOOKS, and the GRAPPLING-HOOKS interpretation is selected. This example is discussed further in §7.6.1.[1]

Very Strong Expectations — Frequency

The second kind of very strong expectations are *frequency expectations.* If two interpretations differ only in the frequency of the last construction which they integrated, and if one of these constructions was much more frequent than the other, the interpretation that integrated this construction will be preferred.

For example, (7.5) causes a garden path reaction in most readers. In the intended interpretation of the sentence, *"complex"* is a noun, and *"houses"* is a verb; thus the students are housed by the complex. However, most readers initially interpret *"the complex houses"* as a noun phrase, and are confused by the lack of a verb.

(7.5) The complex houses married and single students and their families. [2]

The two interpretations do not differ in valence or constituent expectations; the most recent integration of both interpretations fills a constituent expectation. However, these last integrations differ significantly in frequency; the frequency of *"house"* as a verb (according to Francis & Kučera (1982)) is **53** per million [3] , while the frequency of *"house"* as a noun is **662** per million. Because of this order-of-magnitude difference, the nominal sense of house is selected over the verbal sense.

We define a strong frequency expectation as one in which the more frequent construction is at least an order of magnitude more frequent than the alternative. Note that this definition of *strong frequency expectation* is thus similar to the definition of the *access threshold* in the access theory, which allowed a construction to be accessed by evidence *unless* the evidence was more than an order of magnitude more frequent than the construction.

Weak Expectations

Below both strong and very strong expectations on the coherence ranking are interpretations whose last integration fills *weak* expectations. Weak expectations are expectations for *adjuncts*, derived from constraints on the semantic frame associated with a lexical construction. As §3.8.2

[1]Wilensky (personal communication) has also suggested that (7.4b) may be difficult because the nominal sense of *grappling* is very rare, arguing that Sal would prefer (7.4a) because of a very strong *frequency* expectation rather than a very strong *specificity* expectation.

[2]noted by Marti Hearst from an article in the Berkeley campus newspaper.

[3]Or even lower than 53 per million; of these 53 verbal occurences, 29 consist of the gerund *"housing"*, leaving only 24 true verbs.

discussed, the *valence* of lexical constructions expressed their *required* arguments. Optional arguments are represented by slots in the *definition* of the semantic frame associated with a construction. For example, Gawron (1983) suggested that the fact that certain verbs, such as activity verbs, but not others, like statives, are particularly compatible with temporal adjuncts, be represented by including a temporal slot in the definition of the frame related to the verb, but *not* including this slot in the verb's valence arguments. This distinguishes weak expectations from strong expectations, which are specified either in a verb's valence structure or by a construction's constituents. Among the consequences of this representation are that a time adverbial will preferably attach to an *action verb* or an *event noun* over a stative or non-event noun. In general, an active verb is preferred to a stative, and a verbal form is preferred to a nominalization, particularly with deverbal nominalizations from punctual verbs, Thus, for example, when the selection algorithm must choose between an adverbial which could modify a noun, (in this case a deverbal nominalization from a punctual verb), and an activity verb, the activity verb will be chosen, with the result that the preferred interpretation of (7.6a) below is that the talking occurred yesterday, while the preferred interpretation of (7.6b) is that the confirmation occurred yesterday.

(7.6) a. Humbert was talking about Clarence Thomas' confirmation yesterday. *(talking yesterday)*

b. Humbert was talking about Clarence Thomas being confirmed yesterday. *(confirmed yesterday)*

Locality

If none of the interpretations in the interpretation store has recently filled any expectation on the Coherence Ranking, Sal prefers the interpretation which is more *local*. Locality is a simple heuristic which instructs the selection mechanism to choose the interpretation whose most recent integration integrated the *nearest* or *most local* constituents. Thus in sentences like the following, (from Wanner (1980)), where both verbs are compatible with a time adverbial, but neither has an expectation, the selection algorithm will attach the adverb to the nearer verb *"die"*.

(7.7) Bill said John died yesterday.

Locality will be used to account for the some attachment preferences of adverbs and preposition phrases. Note that locality is not a very significant part of the selection algorithm — locality preferences only apply if no coherence criteria are applicable. See §7.5 for further details.

Constraint Violations

Finally, of course, an interpretation can be ruled out because some constraints were violated in building it. The simplest case occurs when an interpretation is ruled out by syntactic constraints. For example, §7.6.1 shows that the word *can*, which can be a MODAL, a NOUN, or a VERB, can be disambiguated by the syntactic constraints of the preceding verb.

An interpretation can be ruled out for violation of *semantic* constraints as well as syntactic ones. For example, Hobbs & Bear (1990) noted examples like (7.8), where the preposition phrase *during the campaign* attaches to the (distant) verb *saw* rather than to the (local) noun *president* because *the president* is not semantically modifiable by a duration adverbial.

(7.8) John saw the president during the campaign.

The next two sections, §7.3 and §7.4, will survey linguistic data on these selection preferences. §7.6 will show that the Selection Choice Principle handles a number of classic cases of ambiguity, while §7.5 reanalyzes the need for a locality principle in selection.

7.3 The Selection Timing Principle

A principle like the Selection Timing Principle, which states *when* selection decisions must be made, is necessary for a theory of sentence processing which attempts to model human behavior. Besides the obvious necessity for some minimal statement of the time of disambiguation, a timing principle can be used to explain the existence of certain *garden-path* effects. For example, the timing principle may sometimes require an interpretation to be selected before enough evidence has come in, causing the interpreter to choose an incorrect interpretation and discard the correct one. In other words, because the interpreter is unable to look ahead in the input for evidence before making a decision (unlike Balaam's ass), it can make the wrong decision. Thus the human sentence interpreter trades completeness for tractability.

The Selection Timing Principle instructs the interpreter to prune interpretations whenever the difference between their ranking and the ranking of the most-favored interpretation is greater than the selection threshold σ. In other words, selection timing is accounted for not by specifying when an interpretation is *selected*, but by specifying when an interpretation is *pruned*. That is, if there are two interpretations in the interpretation store, the selection timing principle explains when the least-favored interpretation is *removed* from the interpretation store. When the store only contains two interpretations, there is no difference between selecting the best interpretation and removing the worst one. If there are more than two interpretations in the store, pruning the worst interpretation will still leave multiple possible interpretations.

This method of specifying selection timing by *pruning* less-favored interpretations according to the *arithmetic difference* between their rankings and the rankings of the most-favored interpretation was proposed by Gibson (1991). In this section, we show that Gibson's method of accounting for selection timing data, which emphasized ranking based on syntactic criteria, can be simply extended to include semantic and constructional criteria.

As we discussed above, Gibson proposed four principles which account for preferences in disambiguation — two of these, the *Property of Thematic Reception* and *Property of Lexical Requirement*, are related to the Coherence Ranking. The *Property of Thematic Reception* assigns a processing load to any structure which does not receive a thematic role, while the *Property of Lexical Requirement* assigns a load to any parse with an unfilled expectation which is filled in some other parse. Sal's preference for interpretations with integrated elements over those with unintegrated elements is a generalization of Thematic Reception, while its preference for interpretations which fill expectations is a generalization of Lexical Requirement.

Specifying selection timing consists of choosing the selection threshold σ in terms of the *Coherence Ranking* of §7.2. We propose that the threshold σ be set at **2** coherence points, where coherence points are assigned to the *Coherence Ranking* as follows:

The Coherence Ranking: (in order of preference)

3 pts Integrations which fill a *very strong* expectation such as one for an exact construction, or for a construction which is extremely *frequent*.

1 pt Integrations which fill a *strong* expectation such as a *valence expectation* or a *constituent expectation*.

1 pt Integrations which fill a *weak expectation*, such as for an optional adjunct or include feature matching rather than feature imposing.

1 pt Integrations which fill no expectations, but which are nonetheless successfully integrated into the interpretation.

0 pts Integrations which fill no expectations, and are not integrated into the interpretation.

Consider an example of pruning caused by a single very strong expectation from the GRAPPLING-HOOK construction discussed in §7.6.1. Recall that Milne (1982) found that sentences like (7.9) cause processing difficulty, because *grappling hooks* is preferably interpreted as a construction:

(7.9) #The grappling hooks on to the enemy ship.

The garden path effect described here by Milne shows that the alternative interpretation must have been discarded at the latest by the time the word *to* is processed, since that word would have indicated that the verbal sense of *hooks* was intended. The Selection Timing Principle in fact predicts that the GRAPPLING-HOOK construction will be selected just after the word *hook* is seen. Figure 7.1 shows the results of the factors which indicate the timing of selection according to the Selection Timing Principle. Because the top interpretation in the figure includes a very strong expectation, i.e., one that specifically mentions the word “hooks”, the bottom interpretation will be pruned.

Pruning can also be caused when one interpretation is preferred over another because of both an unmatched integration as well as an unmatched expectation. For example, in the well known garden-path sentence from Bever (1970), in which the phrase *raced past the barn* is ambiguous between a reduced relative clause and a main verb, the main verb reading is preferred in both of these ways.

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

Figure 7.2 shows the two candidate interpretations at the point just after the word *raced* has been interpreted. Note that the main-verb interpretation of *raced* has two coherence points, while the reduced-relative interpretation has none. First, as Gibson (1991) showed, the phrase *the horse* in the reduced-relative interpretation is not integrated with the rest of the semantics of the sentence, while it is integrated in the main-verb interpretation. Second, the main-verb interpretation fills an expectation for a VERB-PHRASE construction which is not filled by the other one.

As Crain & Steedman (1985) and Altmann & Steedman (1988) point out, in context the interpreter must be able to select the reduced-relative interpretation instead. Altmann & Steedman’s (1988) *principle of referential support* states that “An NP analysis which is referentially supported will be favored over one that is not”. Sal does not model intrasentential or pragmatic information,

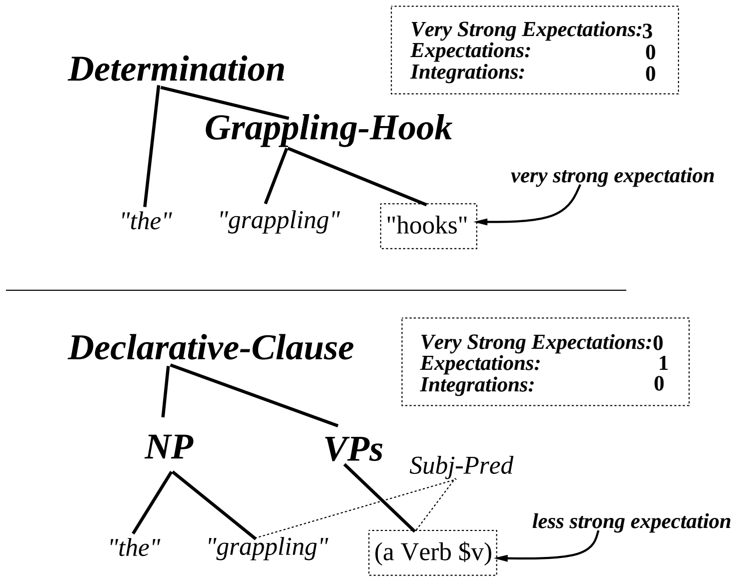


Figure 7.1: Grappling Hooks: The Bottom Interpretation Will Be Pruned

and so is unable to meet this requirement. However, augmenting Sal's integration algorithm with a model of discourse reference such as Hirst's (1986) would enable the current coherence-based selection algorithm to correctly prefer the reduced-relative interpretations in the right contexts, because coreferential noun-phrases will be more coherent than non-coreferential ones.

7.4 Previous Models of Selection

The idea of giving a coherent picture of models of selection and disambiguation is rather daunting. Models vary intensely in their frameworks, their assumptions, the extent to which they are formalized and exactly which problem they attempt to solve. For example a great number of algorithms have been designed quite specifically for a given kind of disambiguation — for example Wilks *et al.* (1985), Dahlgren & McDowell (1986), and Hirst (1984) all propose quite specific algorithms designed to choose between possible sites for the attachment of Preposition Phrases. Because of large amount of related research, I will be somewhat terse in this section. Models of selection *choice* will be discussed followed by models of selection *timing*.

7.4.1 Related Models of Selection Choice

For expository purposes, we divide previous models of selection choice into three groups. The first group, the *coherence* models, are models which are related to and inspired Sal's coherence-based model of selection choice. The second group discusses two other large groups of models, which use respectively *plausibility* and *probability* as their selection metrics. Finally, we discuss

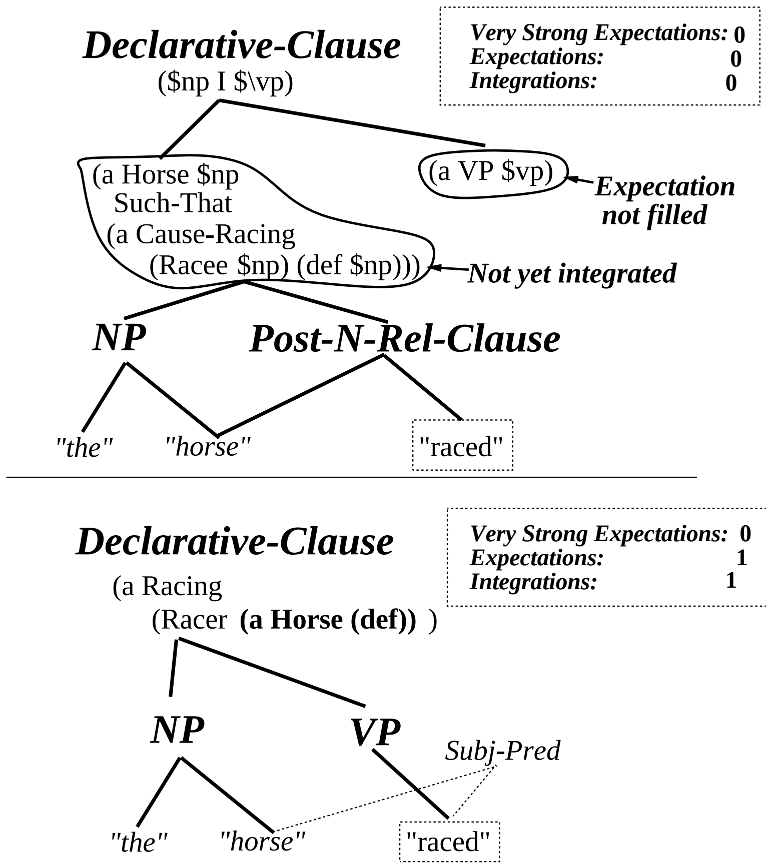


Figure 7.2: The Horse Raced Past the Barn: The Top Interpretation Will Be Pruned

a number of *Syntactic Heuristic Metrics*.

Coherence Models

The idea of using *coherence* as a selection metric was first explicitly stated by Wilks (1975b, 1975a), in describing his model of *Preference Semantics*. Wilks was inspired by what is sometimes known as *the Joos Law* (Joos 1972), which argued for choosing a meaning which was most redundant and hence most coherent with the context (see also Hill (1970) and Joos (1958)).

A number of models have implemented coherence-inspired models by first using *marker-passing* algorithms to find connections between concepts in semantic networks, and then selecting interpretations which were made more coherent by these connections. Such models include Hirst & Charniak (1982), Norvig (1987), and Hirst (1986).

Recently, researchers have proposed that coherence metrics could be used to solve the general problem of *textual abduction* — these include Charniak & Goldman (1988), Ng & Mooney (1990), and Norvig & Wilensky (1990). In a sense, Sal is a special case of many of these more general textual coherence algorithms, in that it concentrates on *local coherence*. Modeling coherence inside the sentence enables Sal to account for the local coherence examples summarized in (7.2) while still allowing the possibility of extending the model to deal with the broader phenomena discussed by the textual models.

Sal's selection theory owes much to the sentence comprehension model of Gibson (1991, 1990a, 1990b). Gibson proposed four principles which account for preferences in disambiguation — two of these, the *Property of Thematic Reception* and *Property of Lexical Requirement*, are related to Sal's Coherence Ranking. The *Property of Thematic Reception* assigns a processing load to any structure which does not receive a thematic role, while the *Property of Lexical Requirement* assigns a load to any parse with an unfilled expectation which is filled in some other parse. Sal's preference for interpretations with integrated elements over those with unintegrated elements is a generalization of Thematic Reception, while its preference for interpretations which fill expectations is a generalization of Lexical Requirement.

Other Global Models

Two other global models of interpretation preferences are popular. The first one was expressed most succinctly by Crain & Steedman (1985:330):

> *The Principle of A Priori Plausibility*. If a reading is more plausible in terms either of general knowledge about the world, or of specific knowledge about the universe of discourse, then, other things being equal, it will be favored over one that is not.

A number of researchers have proposed models along these lines, including Kurtzman (1985), Altmann & Steedman (1988), and Charniak & Goldman (1988). Norvig (1988) has noted two important problems with plausibility-based models. The first, of course, is that it is quite difficult to see how to make them operational. The second is that plausibility models do not explain the cases like the local coherence or garden-paths summarized in (7.2).

The second popular model of interpretation preferences follows Baker (1975/1990) in suggesting that the *probability* of an interpretation be used as a selection metric. A number of parsers have included extensions of standard context-free parsing techniques to *stochastic grammars*, which allows selection to be done based on the relative probability of the candidate parses. For example Fujisaki (1984) and Fujisaki *et al.* (1991) describe an extension to the Cocke-Kasami-Young bottom-up parsing algorithm which uses a grammar in which each rule is augmented with probabilities, producing a final parse tree annotated by a probability measure. Jelinek & Lafferty (1991) show how such probabilities can be computed on-line. Wu (1990) extends this model by proposing that semantic concepts also be associated with probabilities, and that this metric can thus be extended to one for choosing among ambiguous *interpretations*. His method combines the probabilities of both syntactic rules and semantic concepts in assigning a probability to an interpretation in a noun-phrase interpretation task.

The probabilistic models seem quite powerful, and it is possible that they will prove to be extendable to model general selection effects. Some recent extensions of probabilistic models to semantics, like Wu (1990) and (1992), and to general theories of abduction, as described in Hobbs *et al.* (1988) and suggested in Charniak & Goldman (1988) and Norvig & Wilensky (1990), seem quite powerful. Currently, however, the scope of such probabilistic models is still quite limited; they would certainly have to be extended to deal with the local coherence phenomenon summarized in (7.2).

Syntactic Heuristic Metrics

The vast majority of previous models of selection have used a combination of simple syntactic heuristic strategies. This section summarizes the most popular of these strategies.

- Build the Syntactically Simplest Structure
  - Frazier & Fodor (1978) (Minimal Attachment)
  - Wanner (1980) (Arc Ordering)
  - Shieber (1983)
  - Pereira (1985),
  - Kaplan (1972)
  - Cottrell (1985)
- Combine the Closest Structures (discussed further in §7.5).
  - Kimball (1973) (*Right Association*),
  - Frazier & Fodor (1978) (*Local Association*),
  - Frazier (1978) (*Late Closure*),
  - Ford *et al.* (1982) (*Final Arguments*),
  - Schubert (1986) & 1984 (the *Graded Distance Effect*),
  - Hobbs & Bear (1990) (*Attach Low and Parallel*),
  - Gibson (1991) (the *Property of Recency Preference*).
  - Abney (1989) (*[P3] Prefer low attachment.*)
  - Wilks *et al.* (1985)
- Prefer Some Particular Syntactic Categories
  - Ford *et al.* (1982) (*Syntactic Preference*)
  - Abney (1989) (Prefer attachments to verbs over attachments to nonverbs)
- Prefer Arguments to Adjuncts
  - Ford *et al.* (1982)
  - Abney (1989) (*[P1] Prefer argument attachments over nonargument (adjunct) attachments.*)
- Prefer The More Specific Rule
  - Wilensky & Arens (1980)
  - Hobbs & Bear (1990)
  - Charniak & Goldman (1988)

There have been many arguments against simple syntactic heuristics for selection. Many, like Minimal Attachment, are very dependent on particular assumptions about the grammar. Most suffer from their assumption of an autonomous syntax. As many authors have noted (Kurtzman 1985; Norvig 1988; Gibson 1991; Schubert 1986; Osterhout & Swinney 1989), it is quite easy to choose particular lexical items or particular contexts which reverse any of the heuristics.

7.4.2 Previous Selection Timing Models

There are only two major algorithms for deciding *when* to select one linguistic structure over others. The two algorithms are quite similar, and indeed may be notational variants. Both ideas are based on the idea that the best structure should be selected because it is so much better than other structures. The two algorithms differ in whether the top-ranked structure is compared to its *nearest* competitor or to *all* of its competitors, and in whether the comparison is arithmetic or geometric.

The first timing choice model was originally proposed by Luce, and is used in the TRACE model (McClelland & Elman 1986). This model chooses a candidate when the ratio of the activation of the candidate to the activation of all candidates passes a threshold. A structure (in this case a phoneme) is chosen when its response probability passes a threshold of 0.9. The probabilities are determined by using the choice model where the response probability of a given structure (R_i) is its strength divided by the sum of the strengths of all its competitors (where j ranges over the competitors):

$$\text{Pr}(R_i) = \frac{S_i}{\sum_{j=1}^{n} S_j}$$

The second paradigm for selection timing chooses a candidate when the *difference* between the highest candidate and its strongest competitor passes a threshold. This paradigm is assumed by a number of researchers in the lexical recognition domain (Shillcock 1990, Marslen-Wilson 1987, Marslen-Wilson 1990), and by Gibson (1991) for syntactic parsing. Gibson proposed that the top-ranked parse for a sentence is chosen when it differs from the second-ranked parse by a preference factor P.

Bard (1990) suggests that there may not be a great difference between best-competitor and every-competitor models for lexical access models, but her arguments probably would not extend to non-lexical selection models.

Because the implementation of Sal's Selection Timing Principle is an extension of Gibson (1991), the rest of this section will consider his work in further detail.

Gibson proposed that the top-ranked interpretation is chosen when it differs from the second-ranked interpretation by a preference factor P. He expressed this preference factor as a function of the number of *Processing Load Units* associated with local violations of syntactic and thematic criteria. Thus an interpretation which violates less linguistic criteria is preferred, and if the difference between the two preferences is greater than P, the preferred interpretation will be chosen, and the other discarded.

Gibson then showed that this preference factor could be determined empirically by considering two classes of examples. First, a lower limit could be place on P by considering sentences which readers are able to interpret, and yet which have two interpretations which differ by a certain number of PLUs. Next, an upper limit could be placed on P by considering garden paths caused by one interpretation being selected. The different between the two possible interpretations at the point of selection must be greater than P.

Recently Holbrook *et al.* (1988) have proposed a new theory of selection timing, called the *conditional retention theory*. In their model, all meanings of an ambiguous structure are retained until the end of the text, but each meaning is marked with one of three states — *active*, *inactive*, or *retained*. *Active* and *inactive* interpretations correspond to *selected* and *pruned* interpretations, respectively. The novel part of Holbrook *et al.*'s (1988) proposal is the third state, *retention*. An interpretation which is *retained* acts in some ways like a rejected interpretation, in that its meaning does not show facilitation, and yet if later evidence supports that sense, it can be reactivated.

7.5 Principles of Locality in Attachment

Although Sal's selection theory does not place much emphasis on locality, principles of locality or recency play a large role in many models of disambiguation. In many cases, using locality principles to account for certain effects was proposed quite early, and has never been questioned. This section examines much of the data which originally gave rise to theories of locality, and suggests that, although elegant, locality is too simple a principle to account for the broad range of phenomena to which it has been applied.

Since the work of Kimball (1973), a number of scholars have proposed that the human sentence interpreter include some principle like the one Kimball called *Right Association*. Such principles claim that the interpreter should prefer to make attachments to nearby structures. Evidence for such principles includes preferences for the attachment of extraposed relative clauses, attachment of adverbs, and of post-verbal particles. For example, Kimball claimed that (7.11a) could only mean that the job was attractive, not that the woman was attractive. Thus it could not have the same sense as (7.11b).

(7.11) a. The woman took the job that was attractive.

b. The woman that was attractive took the job.

Kimball's argument was that the parser attached the relative clause *that was attractive* to the nearest noun phrase, in this case the noun phrase *the job*, and was unable to attach it to the noun phrase *the woman*.

Principles like Kimball's include Frazier & Fodor (1978) (*Local Association*), Frazier (1978) (*Late Closure*), Ford *et al.* (1982) (*Final Arguments*), Schubert (1986) & 1984 (the *Graded Distance Effect*), Hobbs & Bear (1990) (*Attach Low and Parallel*), and Gibson (1991) (the *Property of Recency Preference*). Each of these "Locality Principles" combine with other processing principles to constrain the actions of their parsers. Three of these models are described specifically enough to test predications about selection.

The first model, the *Local Association* principle of Frazier & Fodor (1978), establishes a fixed-length buffer which can hold five or six words, and predicts that locality effects can be explained by limited view imposed by this buffer. In the next model, the *Recency Preference* principle of Gibson (1991), whenever there is more than one possible attachment point for an adverbial all but the most recent attachment point are removed from consideration. In his case, adverbials can be attached to verbs or sentences, so the Recency Preference principle requires that an adverb cannot "skip" a local verb and attach to a more distant one. The final model assumes what might be called the *Most Recent Semantically Compatible Attachment* principle, first proposed by Wilks *et al.* (1985) (as the *Rule B* algorithm of the CASSEX program), and used also by Hobbs & Bear (1990) and Whittemore *et al.* (1990). These models attempt to choose among attachments by attempting to attach an adverbial to each possible head, starting with the most recent, and moving further left, and selecting the first one that fits semantically.

While it seems uncontroversial that some sort of locality effects must be accounted for, this section argues that these particular locality principles are insufficient. In some cases, the locality effects can be accounted for in the grammar. This is the case, for example, with restrictive relative clause attachment. I argue that distant attachments of relative clauses are ungrammatical, and that relative clause data should not be used in arguing for locality effects.

For other cases, such as verb-particle attachment, that the correct statement of exactly how much material may intervene between a verb and a particle is quite complex, resembling the well-known difficulty of stating the constraints on Heavy-NP Shift. §7.5.3 shows that no current locality principle is sufficient to account for the data, and suggest the direction that a solution might take.

Finally, in the case of adverbial attachment, it seems that some sort of locality principle must be used, but that not only can this be overridden by expectations, but it can also be over-ridden by lexical semantic anomalies. This argues that any locality principle must be the lowest-ranked of any selection criteria, as is suggested by the Coherence Ranking.

7.5.1 Restrictive Relative Clause Attachment

Kimball's arguments for his *Right Association* principle were based on a number of linguistic phenomena. This section begins with Kimball's data on the attachment of relative clauses, and will summarize evidence that the preference for a relative clause to attach to the immediately preceding noun is a *grammatical* fact, and not a processing one. Indeed, before Kimball, it was generally assumed that sentences such as (7.12a,b) (from Hankamer 1973) were simply ungrammatical (talk about transderivational constraints here?):

(7.12) a. *A man$_i$ married my sister who$_i$ had castrated himself.

b. *I gave a kid$_i$ a banana who$_i$ was standing there looking hungry.

But Kimball claimed that sentences like (7.12a,b) must be grammatical, because they were created by the same *Extraposition from NP* transformation that created (7.13b) from (7.13a). Since (Kimball claimed) (7.13b) was grammatical, (7.12a,b) must also be grammatical, and must only be ruled out for performance reasons. Thus the principle of *Right Association* would attach the phrase *who had castrated himself* to the noun *sister* instead of *man* in (7.12b), and thus (7.12) would be grammatical but *unacceptable*.

(7.13) a. The woman that was attractive fell down

b. The woman fell down that was attractive.

It seems quite clear, however, that Kimball is wrong, and (7.13b) is not at all grammatical, at least in my dialect of English and that of my informants. Hence (7.13b) must be starred:

(7.14) *The woman fell down that was attractive.

But (7.14) cannot be ungrammatical because of any locality principle in the parser. We suggest that it is ungrammatical because the POST-NOMINAL RESTRICTIVE RELATIVE CLAUSE construction requires that the relative clause immediately follow the head noun. This it is this *grammatical* requirement which produces this seemingly local effect where there is a noun, and correctly rules out (7.14). (7.15) below are examples of sentences with restrictive relative clauses which are ungrammatical in the restrictive reading (see below for a discussion of non-restrictive relative clauses).

(7.15) a. *The horse$_i$ won the race that$_i$ I bet on.

b. *The book$_i$ is in the corner that$_i$ I bought.

c. *The deer$_i$ have been dying that$_i$ live up north.

Because we claim that the RESTRICTIVE-RELATIVE-CLAUSE construction is grammatical only immediately following the nominal head, it is important to survey some potential counterevidence. First, note that *non-restrictive* relative clauses can appear in final position, especially those cases which resemble Heavy-NP Shift in having a heavy relative clause, as in (7.16).

(7.16) A car pulled up outside the bar which was painted a fiery red.

Here my informants agree that it is the car which was painted red, although the other interpretation is also quite possible. We can see that it is non-restrictive because (7.17), which forces a restrictive reading by making the noun phrase definite, cannot have the interpretation in which the car is painted red.

(7.17) The car pulled up outside the bar which was painted a fiery red.

Note that a nonrestrictive relative clause is almost always required to use the *wh-* pronouns, rather than *that*, and thus (7.18) is interpreted as a restrictive relative clause, making it difficult to get the interpretation in which the car was painted.

(7.18) The car pulled up outside the bar that was painted a fiery red.

A second difficulty with the claim that restrictive relative clauses must immediately follow their heads concerns cases where a head noun is followed by multiple restrictive clauses, as in (7.19) [4]:

(7.19) He buried the cat with the fuzzy tail that got run over (not the one that fell down).

Note here that the head noun *the cat* is followed by two post-modifying restrictive clauses, the first a restrictive prepositional phrase, the second a restrictive relative clause. Of course, postmodifying clauses can be iterated in general (see Quirk *et al.* (1972:1297) for examples). But it appears that postmodifying restrictive clauses must appear before postmodifying non-restrictive clauses. That is, although other post-modifying clauses may appear between a head and a restrictive clause, they must also be restrictive clauses.

[4] this example is from Marti Hearst, personal communication

7.5.2 Adverbial Attachment

One of the most frequently-cited arguments for the locality principles is the frequent association of adverbs and other adverbials with the immediately preceding verbs. For example Kimball (1973) claimed that in (7.20) the sentence-final adverb "yesterday" attaches most easily as a modifier to "rain" rather than to "say" or "expect".

(7.20) Joe said that Martha expected that it would rain yesterday.

Here Kimball's Right Association principle predicts that the adverb *yesterday* will attach to the lowest verb, i.e., the one closest to the adverb. Wanner (1980) shows similar effects for (7.21a), and Gibson (1991) for (7.21b).

(7.21) a. Bill said John died yesterday.

b. Bill thought John died yesterday.

Examples like (7.21) are convincing that between two possible attachments with exactly the same weight, the local one is preferred. However, as many researchers have noted, there are many factors which can cause a more distant interpretation to be preferred. The Coherence Ranking claims that any locality effects must be considered in selection only if no coherence considerations are applicable. A more distant attachment can chosen because of coherence reasons in two ways. First, a distant integration is preferred if it fills an expectation. Second, failing to meet local semantic constraints can rule out a local attachment, causing a distant attachment to be preferred.

Before discussing the data, it is important to note that supporting data must be examined quite carefully; there are unfortunately very few empirical studies which include significant amounts of data on local attachment of adverbs, and some proposed supporting data is confounded with preferences from lexical expectations. For example (7.22) from Schubert (1986) seems suspect, because the verb *leave* may have a *Leaving-Time* argument:

(7.22) John said that he will definitely leave yesterday.

Distant Because of Strong Expectations

Certainly the preference for local attachment can be overridden by valence or constituent expectations, as in (7.23).

(7.23) The woman positioned the dress on that rack.

Distant Because of Constraint Violations

As §7.6.2 showed, certain attachments can be ruled out by constrain violations in the integration algorithm, causing a distant attachment to be preferred to a local one. For example the fact that the time adverbial *every Friday* is compatible with the verb *tell* in (7.24a) but not with the verb *loves* causes a distant attachment to be preferred. Note in (7.24b) that a local attachment is fine if it involves a verb like *wash*, which is compatible with the adverbial.

(7.24) a. Russell tells Regina he loves her every Friday (DISTANT)

b. Russell tells Regina he washes his hair every Friday. (LOCAL)

Similarly, Hobbs & Bear (1990) noted examples like (7.25), where the preposition phrase attaches to the distant verb rather than the local noun because *the president* is not semantically modifiable by a duration adverbial.

(7.25) John saw the president during the campaign.

Hobbs & Bear note that the non-local attachment in (7.25) cannot be due to a syntactic preference for verbs alone, because an event noun such as *demonstrations* is acceptable as a head for a duration adverbial. Thus in (7.26), most readers interpret the demonstrations as having taken place during Gorbachev's visit:

(7.26) The historian described the demonstrations during Gorbachev's visit.

7.5.3 Verb-Particle Attachment

Another kind of phenomenon which is commonly cited as evidence for locality principles is the attachment of verbal particles to their head verbs. Kimball (1973) claims that Right Association explains why (7.27a) is unacceptable, and why (7.27b) cannot be interpreted such that the main verb of the sentence is *figure out*:

(7.27) a. Joe figured that Susan wanted to take the train to New York out.

b. Joe figured that Susan wanted to take the cat out.

In (7.27a), Kimball's Right Association principle predicts that readers should have difficulty associating the particle *out* with the verb *figure*, since they are so far apart. Similarly, in (7.27b), the locality principle causes the reader to attach the particle *out* to the verb *take*. More recent versions of locality principles make similar claims, which I will describe below.

As with the other linguistic phenomena cited in previous sections, particle attachment data does not sufficiently support any of the locality principles in the literature. This section will show that the correct statement of exactly how much material may intervene between a verb and a particle is quite complex, resembling the well-known difficulty of stating the constraints on Heavy-NP Shift. Because of this complexity, the two most well-defined locality principles cannot account for the particle attachment data. The rest of this section will discuss these two principles, Frazier & Fodor's (1978) *Local Association* and Gibson's (1991) *Recency Preference*, and then sketch the direction that a solution to the verb-particle problem might take.

As in the case of restrictive-relative-clause attachment, linguists before Bever (1970) generally assumed that the constraints on exactly what sort of objects phrasal verbs could take, and the relative positioning of the objects and the particles, were grammatical ones.

Fraser (1976) notes that no locality principle which is stated specifically in terms of *number of words*, such as Local Association Principle of Frazier & Fodor (1978), can account for the verb-particle data. He notes that (7.28a), which includes a four-word noun phrase between the verb and particle, is uninterpretable. But (7.28b)–(7.28d), which include interrupting noun phrases with *five* words, are interpretable. Thus whatever the constraints may be on the placement of verb-particle objects, they are not statable in terms of constituent length.

(7.28) a. *I called *the man who left* up

b. He called *all of my best friends* up.

c. Won't you total *some of those larger figures* up.

d. Some charged *the adding machine fire-loss* off to experience.

The most explicit principle is Gibson's (1991) Recency Preference principle, which states that whenever there are more than one possible attachment points for an element, all but the most recent one are removed from consideration. Thus attachments are always made to the most recent element, as in the examples in (7.29):

(7.29) a. Bill thought John died yesterday.

b. John figured that Sue took the cat out.

Gibson's principle is an improvement over Kimball and Frazier & Fodor, as it handles the majority of the particle attachment data. His principle nonetheless seems insufficient to deal with all attachment data. In particular, there are cases of infelicitous attachments which are not predicted by his theory. That is, there are times when a particle attachment is uninterpretable even if there is no possible intervening attachment point. For example, *Recency Preference* would predict that a very long noun phrase without an embedded verb phrase should be interpretable, as there are no attachment points for verbal particles. However, (7.30a)–(7.30c) have no embedded verbs and yet are uninterpretable:

(7.30) a. *He threw the rotten apple from the tree behind our house out.

b. *I wrote that tedious problem set due Monday up.

c. *I called my friend, the one from New York, up.

Notice that in examples (7.30a)–(7.30c) complex noun phrases intervene between the verb and the particle. Notice also that each of these noun phrases is post-modified. In (7.30a), the noun head *apple* is modified by two post-nominal prepositional phrases. In (7.30b), the head *problem set* is modified by a postnominal adjective-phrase, while in (7.30c) the head *friend* is post-modified by a nonrestrictive appositional noun-phrase.

Besides all the examples in (7.30), all of Gibson's examples are postmodified — as might be expected for noun phrases which include verbs. So it seems like rather than ruling out intervening verbs, the construction rules out intervening units which are too complex. In all the example above, and in Gibson's data, the intervening elements were all more than one clause or intonation group. Verb-particle attachments thus cannot be ruled out simply because an alternative attachment intervenes. Constraints on attachment must be expressed in terms of particular properties of the intervening noun phrase (such as its complexity, or the number of intonation groups) which make it an unsuitable candidate for the verb-particle construction.

One alternative proposal is that the problem with sentences like (7.30) and (7.29) is not that they are ruled out in selection, but that they are unable to be accessed. For example, we might suppose that the verb-particle construction can only be accessed once both constituents have been seen. If this is the case, we might propose that if a phrase which is too complex intervenes between the verb and particle, the second constituent is not recognized as part of the same construction as the first. Thus the verb-particle construction is never accessed, and the the interpreter would assume that the verb is a bare-verb, and not a constituent of the verb-particle construction. We can test this hypothesis by looking at verb-particle combinations in which the bare verb cannot occur without the particle, or in which the verb without the particle has very different subcategorizations. In these cases there will be no ambiguity — the verb will be recognizable as a constituent in the VERB-PARTICLE construction. For example, while there is a verb *cordon off*, there is no verb *cordon*. Thus appearance of *cordon* as a verb ought to be evidence for *cordon off*. In fact, it is interesting to note that such examples, like (7.31) below, are slightly better than the examples above.

(7.31) a. ?They cordoned the ramp which led to the ship off.

b. ?The cop pulled the driver who had just sped by over.

c. ?The teacher singled the kid who had just emigrated from Korea out for special attention.

Although 7.31a-c) are somewhat better than 7.30a-c), they are nevertheless bad. This casts doubt on the hypothesis that distant constituents keep the VERB-PARTICLE construction from even being *accessed*. The fact that there is no other reading may give some weight to the cordon-off construction, but some sort of *grammaticality* constraint seems to still rule it out.

In conclusion, we suggest that the VERB-PARTICLE construction be represented in the grammar as a distinct construction, with distinct semantics (Makkai 1972; Fraser 1976; Bolinger 1971). In addition, the *grammar* would include principles about what sort of material could intervene between constituents in such multi-constituent lexical constructions (perhaps including the 'S GENITIVE construction). These requirements might be specified in terms of *intonational units* or perhaps number of *focus points*.

7.6 Testing the Selection Choice Principle

This section presents a number of classic cases of local ambiguity, showing that the selection choice principle is sufficient to choose among them.

7.6.1 Lexical Ambiguity

Constraint Violations

Sal can disambiguate lexical ambiguities in a number of ways. The simplest of these is when one possible interpretation is ruled out because it fails to meet constraints during integration. For example, Figures 4.5 and 4.6 in §4.5.2 showed how the integration algorithm disambiguated the word *can* (which can be a MODAL, a NOUN, or a VERB) in the sentence fragment (7.32):

(7.32) Peter will can . . .

Recall that (7.32) might be completed as in (7.33a) or (7.33b):

(7.33) a. Peter will can all this salmon by 5:00.

b. Peter will can that employee who was accused of insider trading.

The system was able to rule out the nominal and auxiliary senses of the word *can* by examining the sentential context. The sentential context requires a verb (because of the constraints the auxiliary *will* places on its complement) and hence only of the verbal sense of *can* is allowed.

The system was able to do this disambiguation because syntactic and semantic information are all available on-line; without verbal subcategorization information, it would be impossible to disambiguate (7.32). Resolving this sort of ambiguity is quite simple, given the necessary knowledge. More complex cases of lexical disambiguation occur when both interpretations are syntactically felicitous.

Very Strong Expectations — Specificity

Very strong expectations can be very strong because they are for specific constituents, or because they are for very frequency constituents. This section discusses the former case. §7.2 mentioned the problem of disambiguating the phrase *grappling hooks* in (7.34) from Milne (1982) in which the word *hooks* can function as a noun (as in (7.34a)) or a verb (as in (7.34b)):

(7.34) a. The grappling hooks were lying on deck.

b. #The grappling hooks on to the enemy ship.

Recall that the use of *hooks* as a noun, as in (7.34a), is much preferred, and that Milne (1982) found that sentences like (7.34b) cause processing difficulty. The preference for (7.34a) is accounted for by the Selection Choice Principle because of the existence of the GRAPPLING-HOOKS construction. Because the second constituent of the construction is constrained to be the word *"hook"*, the construction has a *very strong* expectation for *"hooks"*. In (7.34b), on the other hand, the DECLARATIVE-CLAUSE construction only gives rise to an expectation for a VERB — ie for any verb. This expectation is not a very specific one. Figure 7.3 shows the two candidate interpretations. Because the expectation from the GRAPPLING-HOOKS construction is stronger than the expectation from the DECLARATIVE-CLAUSE construction, the GRAPPLING-HOOKS interpretation is selected.

Hobbs & Bear (1990) note that the complementizer interpretation of the word *that* is preferred to the determiner interpretation, all things being equal. For example in the preferred interpretation of (7.35), *that* is a complementizer beginning the SUBORDINATE-PROPOSITION construction, rather than a demonstrative determiner of the noun *"sugar"*.

(7.35) I know that sugar is expensive.

Again, the Selection Choice Principle accounts for this preference, because the SUBORDINATE-PROPOSITION construction specifically requires the word *that* as a constituent. The NOUN-PHRASE construction, which is the next constituent, *may* begin with the demonstrative determiner *that*, but it is not required. The two possible interpretations just after seeing the word *know*, (and before seeing the word *that*) are shown in Figure 7.4.

The first interpretation in Figure 7.4 has a very strong expectation for "that" (i.e., one that is specific to the word "that") which is worth 3 points, while the second interpretation only has a strong expectation (in this case for the more general construction NP), which is worth 1 point, so the first interpretation is preferred.

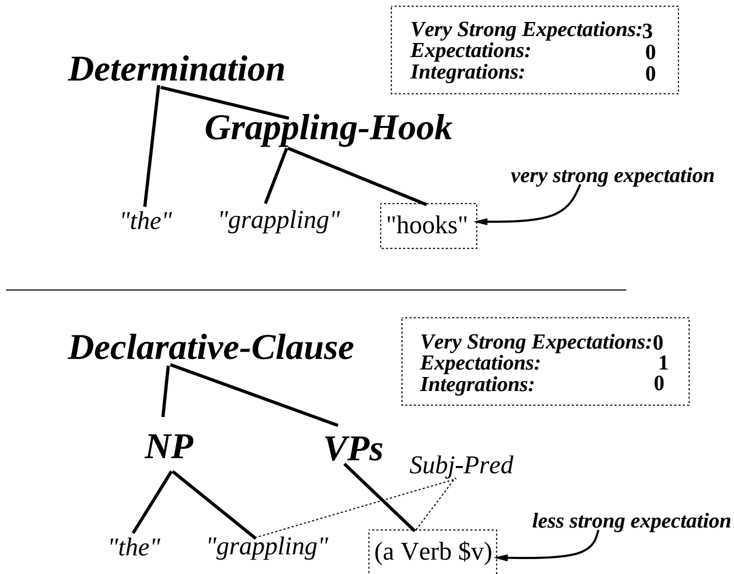


Figure 7.3: Two Interpretations before seeing "hooks"

Very Strong Expectations — Frequency

Lexical disambiguation also can rely on the frequency of the construction which was last integrated. For example, §7.2 noted that (7.36) (a repeat of (7.5) above) causes a garden path reaction in most readers, because the intended interpretation requires "houses" to be interpreted as a verb, a very infrequent usage.

(7.36) The complex houses married and single students and their families.

We noted that the frequency of *"house"* as a verb (according to Francis & Kučera (1982)) is **53** per million , while the frequency of *"house"* as a noun is **662** per million. Because of this order-of-magnitude difference, the nominal sense of house is preferred to the verbal sense. The two interpretations of the phrase *"the complex houses"* are shown in Figure 7.5. Note that the nominal interpretation fills a very strong frequency expectation, yielding 3 points, while the verbal interpretation only fills a strong expectation, yielding 1 point. The difference, 2 points, passes the selection threshold, causing the verbal interpretation to be pruned, and the nominal interpretation to be selected.

7.6.2 Adverb and Preposition Attachment

We turn now from lexical construction ambiguity to ambiguity among non-lexical constructions. This sort of ambiguity is often called "structural" ambiguity and is often treated differently than lexical ambiguity. Following the **Uniformity Principles** of Chapters 3 and 4, the selection

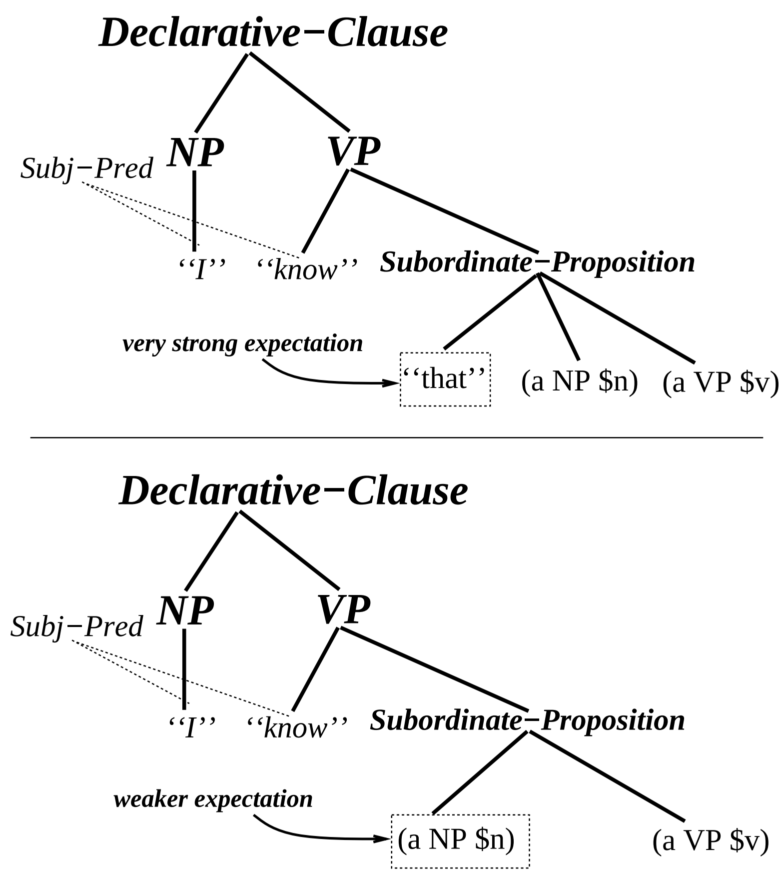


Figure 7.4: Two Interpretations before seeing "that"

algorithm treats lexical, structural, and other ambiguities uniformly. Each is treated as a choice between interpretations based on the construction which was most recently integrated.

The most frequently-discussed type of non-lexical ambiguity is generally called *attachment* ambiguity and refers to the ambiguity caused by the fact that prepositional and adverbial phrases may possibly modify one of various heads in a sentence. A great number of models have been proposed to explain the preference for these attachments. Some of these algorithms have been specific to preposition-phrases – like Wilks *et al.* (1985) and Dahlgren & McDowell (1986). Others have been more general, but have attempted to give completely syntactic solutions to the attachment problem (like Frazier & Fodor 1978, Ford *et al.* 1982, and Hobbs & Bear 1990).

This section shows that the Selection Choice Principle accounts for the disambiguation of various constructions involving preposition-phrases and adverb-phrases in a more general way than previous solutions. It is more general in accounting for "attachment" ambiguities than models which give specific procedures for preposition-phrases, or those which account for only syntactic effects.

As is true with ambiguities in general, attachment ambiguities fall into three classes — those

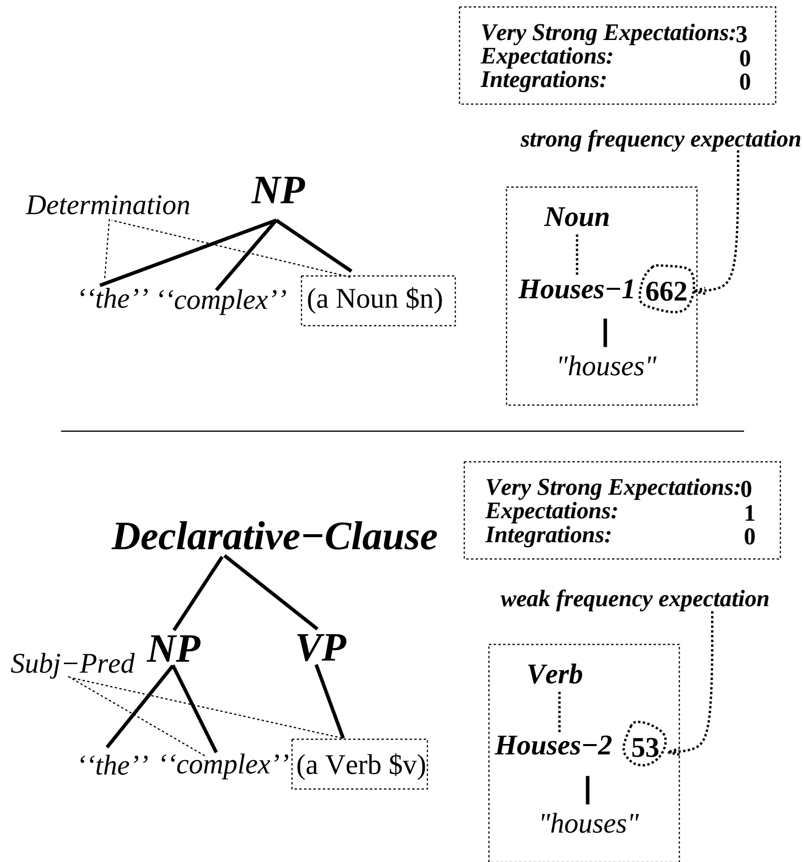


Figure 7.5: Two Interpretations before seeing "houses"

where one of the choices involves an expectation, those where both do, and those where neither do:

1. Choosing between using a prepositional phrase to fill some previous expectation (such as a nominal or verbal valence role, or an expected constituent of a construction) or accessing and filling a new construction (such as a Postnominal-Modifier-PP, or a Postverbal-Modifier)

2. Choosing between filling two previous expectations (such as between a nominal and verbal valence expectation).

3. Choosing between filling two new constructions, neither of which involve previous expectations. For example, choosing between a verb-phrase modifier such as in the Benefactive Construction, and a noun-phrase modifier like the Postnominal-Modifier-PP construction.

The rest of this section will give a number of examples of each of these types of choices. A great number of preferences are shown to fall out of situations like (1) above, which indeed is a major design constraint on constructions, while a smaller number of examples are given for situations (2) and (3). In general, although the Selection Choice Principle is expressed in an elegant and general way, the actual application of the principle involves representing quite specific knowledge about different kinds of verbs and prepositions. Although this section only works through a small number of examples, the point is clear: attempted solutions to disambiguation which do not involve detailed and exhaustive study of individual lexical semantics are unlikely to prove generally successful.

The section will discuss these examples in the following subsections: *Verbal Valence Expectations, Nominal Valence Expectations, Weak Expectations, The Benefactive Construction versus Post-Nominal-PP,* and *Attachments Which Violate Constraints.*

Verbal Valence Expectations

The most obvious corollary of the Selection Choice Principle is a preference for verbal arguments over any adjuncts. For example, Ford *et al.* (1982) showed that readers of (7.37) preferred to interpret the phrase *on that rack* as a complement of the verb *position*, rather than as a modifier of the noun *dress*. We assume following Ford *et al.* that the verb *position* has two thematic frames, one of which has a valence position for a location.

(7.37) The woman positioned the dress on that rack.

After processing the beginning of (7.37), the three arguments of the trivalent sense of the verb *position* are filled as follows:

(7.38) Positioning-Action $p
    Positioner **(a woman $w)**
    Positioned **(a dress $d)**
    Position-Location **$x**

Note that the third argument of the verb, the **Position-Location**, is currently empty. After processing the last preposition-phrase, the interpretation store will contain the following two candidate interpretations:

(7.39) Positioning-Action $p
    Positioner **(a woman $w)**
    Positioned **(a dress $d)**
    Position-Location **(a location (On $p))**

(7.40) Positioning-Action $p
    Positioner **(a woman $w)**
    Positioned **(a dress $d**
        (located-on
            (a rack $r)
            ($d)))
    Position-Location **$x**

(7.39) will be preferred over (7.40) by the Coherence Ranking because the last integration filled a valence expectation. (7.39) will be assigned 2 coherence points, one because the last integration filled this expectation, and one because it fit into the current interpretation. (7.40 will be assigned only 1 coherence point because the last integration fit successfully into the interpretation.

Nominal Valence Expectations

Because Construction Grammars allow any lexical construction to have valence, valence expectations are associated with nouns as well as verbs. This section summarizes data where an interpretation where a preposition phrase fulfills a nominal valence expectation is preferred to one in which a preposition phrase acts as a post-verbal-modifier.

Taraban & McClelland (1988) studied the role of expectations in a number of preposition-attachment ambiguities. They studied expectations that were generated when a sentence had been processed up to and including the preposition, but not including the prepositional-object head noun. In general, they found that subjects used both verbal and nominal valence expectations to try to attach the prepositional objects.

Taraban & McClelland did not attempt to provide an algorithm for choosing between nominal and verbal attachment. But an examination of their data shows that a great percentage of the sentences in which subjects preferred a noun-phrase attachment over a verb-phrase attachment can be accounted for by considering the constraints that certain nouns put on their arguments.

Examples (7.41)–(7.43) from Taraban & McClelland show a final word which was in line with subjects expectations (noun phrase attachment) as well as one which was unexpected (verb phrase attachment). Subjects' expectations for noun-phrase attachments were quite strong by the time the final noun was read, causing sentences with the first choice of final word to be read much quicker than those completed with the second choice.

(7.41) The executive announced the reductions in the *budget / evening*.

(7.42) The philanthropist appreciated the story on his *generosity / deathbed*.

(7.43) The high-school senior stated his goals for the *future / principle*.

Note that in each of these cases the prepositional phrases fill a nominal valence slot. Deverbal nominalizations like *reductions* have valence slots like the related verbs (in this case for a *Reducer* and a *Reduced*), while nouns like *story*, *report*, or *book* which describe written documents have valence slots for the *Content* of the documents.

Examples of nominal valence occur also in the data of Whittemore *et al.* (1990), who noted that noun-phrase attachment always occurred with *partitive nouns* in their data. They include examples such as (7.44):

(7.44) a. the legs of your trip.

b. the size of the hotel.

Hirst (1986) gives another nominal-valence example (his 6-118) where the phrase *sexual intercourse* includes a nominal valence for a prepositional phrase headed by *between*.

(7.45) One witness told the commissioners that she had seen sexual intercourse taking place between two parked cars in front of her house.

Weak Expectations

The weak expectations discussed in §7.2 allows the semantics of certain verbs or nouns to be specified in the *definitional language* of §3.8.1 as expecting certain kinds of adjuncts. Because these are not specified in the *constitute* of the construction, they are not valence expectations, and thus are weak rather than strong. Among the consequences of this representation are that a time adverbial will preferably attach to an *action verb* or an *event noun* over a stative or non-event noun. In general, an active verb is preferred to a stative, and a verbal form is preferred to a nominalization, particularly with deverbal nominalizations from punctual verbs, which, as Quirk *et al.* (1972:1290) point out, "might be described as mere records of an action having taken place rather than as descriptions of the action itself". Similarly location adverbials or preposition-phrases will prefer locative-stative verbs.

For example, Ford *et al.* (1982) present experimental results on a number of examples of adverb attachment where readers did not choose the most local attachment for time adverbials. These include (7.46a)–(7.46c), where half the readers choose the local attachment of the time-adverbial, and half choose the distant attachment.

(7.46) a. The men discussed John's killing himself last night.

b. Tom discussed Bill's dying yesterday.

c. The teachers discussed our selling the drugs yesterday.

Note that these are all nominalizations, while the main verb *discuss* is an active verb. Compare these to cases like (7.47) below, where the preference is for attaching *yesterday* to *died*. First, the embedded verb here is active, and in addition the semantic frame for the main verb *though* seems less likely to have a slot for a *time* argument than the verb *discuss* above.

(7.47) Tom thought Bill died yesterday.

Similarly the preferred interpretation of (7.48a) below is that the talking occurred yesterday, while the preferred interpretation of (7.48b) was that the confirmation occurred yesterday.

(7.48) a. Humbert was talking about Clarence Thomas' approval yesterday. (DISTANT)

b. Humbert was talking about Clarence Thomas being approved yesterday. (LOCAL)

Similarly, all else being equal a locative preposition-phrase will attach to an action verb rather than a noun-phrase. This is because the semantic frames for actions are marked for location, where the concepts for nouns like "man" are not. It is *possible* to integrate locations with nouns, but it is not *expected*.

(7.49) a. He shot the squirrel in the park.

b. I came upon that little house in the park.

The Benefactive Construction versus Post-Nominal-PP

This section discusses the disambiguation of a certain class of preposition-phrases headed by *for*. Many uses of *for* can be disambiguated by looking at the argument of the preposition; This is the case for time adverbials like "for three years". Like other time adverbials, these uses of *for* preferably attach to verbs and event nouns, for the reasons discussed above.

This section is limited to the discussion of one particular sense of *for*, the sense commonly called the *benefactive* sense. There is a preference for prepositional-phrases dominated by the benefactive *for* to modify verbs rather than nouns, especially volitional accomplishment verbs.

For example, Ford *et al.* (1982) found that readers preferred the verb-phrase attachment in (7.50a), and Gibson (1991) claimed the same for (7.50b), in which the verbs are volitional accomplishment verbs, while (7.51 shows a number of cases in which readers prefer noun-phrase attachments when the verbs are stative or non-accomplishments.

(7.50) a. Joe carried the package for Susan.

b. The woman wanted the dress for Mary.

c. I bought the flowers for the children.

(7.51) a. Joe included the package for Susan.

b. That book is the present for Mary.

Clearly this preference cannot be stated as a generalization about preposition-phrases. That is, this preference is a fact about *for*, not a fact about all preposition-phrases.

One way of accounting for the preference is to claim that verbs like *carry*, *want*, and *buy* have a thematic grid which has an optional benefactive argument. If this is the case, the preference for verbal attachment of these benefactive phrases would fall out of the Selection Choice Principle, because their would be a verbal valence expectation for them.

The problem with this claim is that the use of these *for*-phrases is quite productive — they can be used with a great number of verbs, and as new verbs are created, the new verbs can be used with the benefactive as well. Theories of grammar like LFG can account for this by proposing *lexical rules* which allow an extra valence position to be added to verbs — such a rule might add a benefactive valence argument to the verb *carry*, and to any verb which falls into a certain equivalence class. Such lexical rules, operating on the semantic structures of verbs, have been proposed by Pinker (1989).

Lexical rules, however, are not used in CIG. Recall that the use of lexical rules is ruled out by the Interpretive Hypothesis of Chapter 2. As Chapter 2 discussed, capturing some generalization with a lexical rule causes the grammar which is used for capturing generalizations to be distinct from the grammar which is used by the interpreter. Goldberg (1991) makes a number of other arguments against such lexical rules.

Instead of proposing lexical rules and capturing the use of the benefactive *for*-phrase as a valence argument, CIG includes a *construction*, the BENEFACTIVE construction. Like the DATIVE construction of Jurafsky (1988) and the DITRANSITIVE construction of Goldberg (1989), the BENEFACTIVE construction has constituents for a verb as well as for its complements. The construction requires a complement which is a preposition-phrase headed by *for*, and places strict requirements on the kinds of verbs that it can combine with.

Consider the kind of verbs that can felicitously combine with the construction. Note first that statives are unacceptable; each of the sentences in (7.52) is ungrammatical because the verbs cannot combine with the benefactive, while the nominal attachment is ruled out because the pronouns are not modifiable.

(7.52) a. *I have it for Mary

b. *The box contains them for Mary.

The constraint seems to be that the verb involves somebody doing something which causes some resultant state or event — i.e., accomplishment verbs, in the Vendler scheme, or more specifically volitional accomplishments (so as to rule out **They recovered from illness for Mary*, at least in the non-volitional reading).

By defining the BENEFACTIVE construction, we have shown why examples like (7.52) are ungrammatical and why nominal attachment is preferred in sentences like (7.51). It remains to be explained why the BENEFACTIVE construction itself is always chosen over the POSTNOMINAL-PP-MODIFIER construction. We argue that this selection preference is caused by *coherence*. Note that the BENEFACTIVE construction places very specific constraints on its head — that it be an accomplishment verb, indeed a volitional accomplishment verb, and has a lexical constraint for the preposition *for*. The POSTNOMINAL-PP-MODIFIER construction, on the other hand, has very general constraints — it constrains its head to be a NOUN and its preposition to be any subclass of PREPOSITION. Thus the BENEFACTIVE construction has more specific expectations for its constituents, thus being more *coherent* than the other construction, and hence all things being equal, will be preferred by the coherence ranking.

Of course as Ford *et al.* (1982) showed with (7.53), it is possible to change the preferences for the benefactive by manipulating the expectations set up by the context.

(7.53) a. When he arrived at our doorstep, I could see that Joe carried a package for Susan. *(the package is for Susan)*

b. Whenever she got tired, Joe carried a package for Susan. *(the carrying was for Susan)*

Attachments Which Violate Constraints

Of course many cases of local ambiguity can be resolved quite simply because one of the interpretations violates semantic constraints, and thus fails to integrate. For example, the fact that locative adverbs cannot modify stative verbs (of course excepting certain locative statives such as *stand*), accounts for the attachment preferences in (7.54) from Gibson (1991). Gibson (1991) argues that the attachment preferences in these sentences should be accounted for instead by a locality preference. However note that when the preposition phrases are forced to attach to the verbs, in (7.55), the sentences become ungrammatical, indicating a semantic infelicity with the verb attachment.

(7.54) a. John loved the woman in the park.

b. I knew the woman in the kitchen.

c. John believed the secretary at the office.

(7.55) a. *Did John love her in the park?

b. *Where did John believe the secretary?

Another example of an attachment which is ruled out because of the violation of semantic constraints are temporal adverbs. Hobbs & Bear (1990) noted examples like (7.56), where the preposition phrase attaches to the distant verb rather than the local noun because *the president* is not semantically modifiable by a duration adverbial.

(7.56) John saw the president during the campaign.

7.6.3 Adjectives as Modifiers versus Heads

A classic case of ambiguity which can be either lexical or constructional concerns the sentence fragments in (7.57), which are well-known to cause the garden path effect.

(7.57) a. The old man the boats.

b. The prime number few.

The only coherent interpretation of (7.57a) is the one in which *man* is a verb, and the subject of the sentence is the ADJECTIVE-HEADED-NP construction *the old*. But readers do not get this interpretation; they take *the old man* as an NP. Similarly in (7.57b), the only coherent interpretation has *number* as a verb, and *the prime* as the subject, but readers interpret *prime number* as an NP.

As Gibson (1991) points out, the ADJECTIVE-HEADED-NP construction in (7.57a), (whose exemplars also include *the brave*, *the weak*, and *the underprivileged*) is rare, certainly more so than the ADJECTIVE-NOUN construction. More significantly, the ambiguous word *man* has a significant frequency imbalance. Francis & Kučera (1982) show the frequency of the noun *man* to be **2110**, compared with **18** for the verb. Figure 7.6 shows that the Selection Choice Principle will choose the nominal sense of *man*, because the interpretations differ only in this one expectation.

Gibson notes that by choosing a word such as *feed* which has the opposite preference, (the verb form occurs 132 times to the noun's 65), a sentence like (7.58a) which has the same structure as (7.57a) will not cause the garden path effect. Gibson also notes that the use of *feed* as a noun in the same context, as in (7.58b), also does not cause a garden path.

(7.58) a. The old feed the young.

b. The old feed made the horse sick.

The rarity of the ADJECTIVE-NOUN construction seems to neutralize the preference for a verbal interpretation of *feed*, causing both interpretations to remain available.

There are two possible causes for the garden path effect in (7.57a). The first is the frequency difference between the nominal and verbal senses of *man*, while the second is the possibility that *old man* is a collocation. If the latter is the case, the readers' inability to interpret *man* as a verb might be due to the *very strong* expectation from the collocation. Testing this hypothesis requires an adjective which is coherent with the ADJECTIVE-HEADED-NP construction, but which does not form a collocation with *man*. The adjective *underprivileged* fits these requirements, as in (7.59) below:

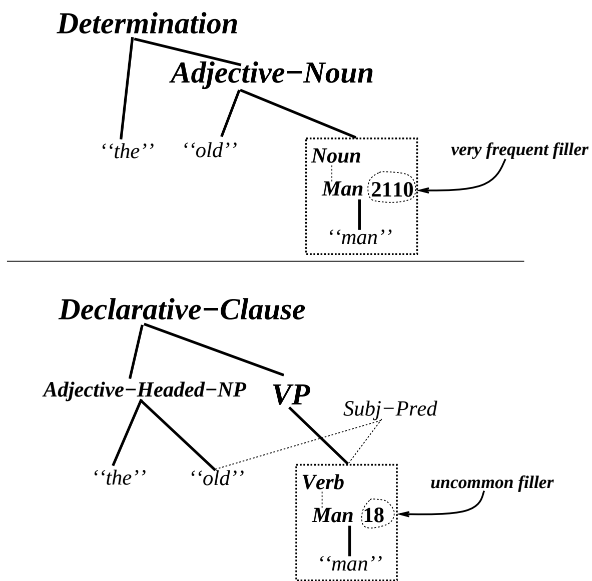


Figure 7.6: Frequency Preferences in the Adjective-Headed-NP construction

(7.59) #The underprivileged man the oars of society.

(7.59) seems to be a garden path sentence, and the possibility of *underprivileged man* being a collocation seems quite remote, as *underprivileged* has a frequency of **3** in the Brown Corpus (Francis & Kučera 1982). Thus the garden path effect must be due to the different in frequencies of the nominal and verbal senses of the word *man*.

A similar phenomenon seems to occur in (7.57b), repeated as (7.60b) below. While (7.60a-b) both cause processing difficulties, Milne (1982) found that (7.60a) was much worse than (7.60a). We certainly expect both of these to be difficult to process, since the frequency of the verbal sense of *number* is only **18**, while the frequency of the nominal sense is **658**. But why should the first example be so much worse than the second?

(7.60) a. The prime number few.

b. The bold number few.

We claim that (7.60a) is worse than (7.60b) for two reasons. The first, mentioned in §7.6.1, is that the phrase *prime number* is a construction in its own right. Milne suggested this factor as well. But a second factor is a semantic one. Quirk *et al.* (1972) Section 7.23 note, the ADJECTIVE-HEADED NOUN construction allows a small class of adjectives which describe classes of *people*, and cannot be extended to any adjective like *prime*. This construction is quite easily described in CIG because CIG allows semantic constraints on constituents — the adjective constituent would simply be constrained to a certain semantic class of adjectives.

7.6.4 Extraposition versus Pronominal *It*

> *"I thought you did," said the Mouse. "I proceed. 'Edwin and Morcar, the earls of Mercia and Northumbria, declared for him; and even Stigand, the patriotic archbishop of Canterbury, found it advisable — "'*
>
> *"Found* what?*" said the Duck.*
>
> *"Found* it,*" the Mouse replied rather crossly; "of course you know what 'it' means."*
>
> *"I know what 'it' means well enough when I find a thing," said the Duck; "it's generally a frog or a worm. The question is, What did the archbishop find?"*
>
> *— Lewis Carroll, Alice's Adventures in Wonderland*

The common ambiguity displayed in this citation from Lewis Carroll can also be resolved by the Selection Choice Principle. The ambiguity revolves around the word *it*: is it to be taken as the beginning of the EXTRAPOSITION construction, or as a normal pronoun?

Crain & Steedman (1985) noted that when processing extraposed clauses such as (7.61), people prefer to analyze the clause *John wanted to visit the lab* as a complement clause rather than as a relative clause modifying *the child*.

(7.61) It frightened the child that John wanted to visit the lab.

Crain and Steedman note that this preference for the complement interpretation can be modified by the context. For example, in (7.62), the context causes the word *it* to be interpreted as a pronoun rather than the start of a extraposition construction. Since the extraposition interpretation is no longer possible, readers interpret the final clause as a relative clause modifying *the child*.

(7.62) *Context:* There was an explosion
It frightened the child that John wanted to visit the lab.

Although the interpreter presented here cannot model the effects of context since it is a single-sentence interpreter, it can model the intra-sentential preferences in processing (7.61). Additionally, some simple assumptions about inter-sentential processing will allow it to model the processing of (7.62).

We begin with (7.61). Figure 7.7 shows the two candidate interpretations of the sentence just after processing the words *It frightened the child*.

There are two candidate interpretations at this point, one involving the extraposition construction, and the other the declarative-sentence construction (thus in the second interpretation the word *it* acts as a normal pronoun). The second candidate interpretation has no expectations, as all the verbal and constructional expectations are already filled. In the first interpretation, however, the SUBJECT-EXTRAPOSITION construction has one unfilled constituent slot — the slot for a SUBORDINATE-PROPOSITION. Thus this interpretation has an expectation for a SUBORDINATE-PROPOSITION, and thus for the word *that* which begins the construction.

When the next word (*"that"*) is processed, the two interpretations appear as in Figure 7.8. The score for the first interpretation will be higher because it contains two fulfilled expectations — first, the constituent expectation from the extraposition construction, and second, the *Frightener*

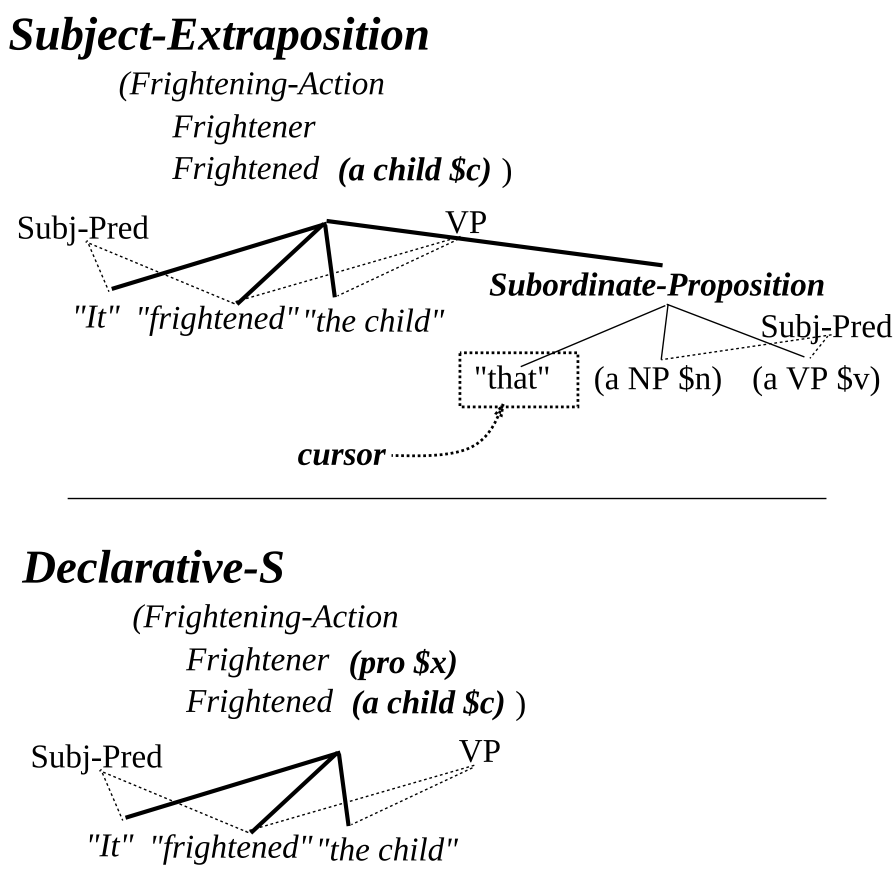


Figure 7.7: Processing an Extraposition (1): The Interpretation Store after *child*

role from the verb *frighten* can now begin to be filled by a *Subordinate-Proposition* variable. The first expectation, the constituent expectation, is a very strong expectation (for the word “that”) and is valued at 3 points. The second expectation is valued at 1 point. Since this integration fits into a top-level interpretation, for another point, the first interpretation achieves a total of 5 points. The second interpretation scores 1 point for fitting into the top-level interpretation, but no other points. The difference, 4 points, is well above the selection threshold of 2 points, causing the second interpretation to be pruned, and the first one selected.

In example (7.62) above, Crain and Steedman showed that readers make exactly the opposite choice in a context which gives an immediate antecedent for the pronoun *it*. Although this interpreter does not handle multi-sentential input, we might nonetheless suggest how this example might be handled. Because the pronoun receives an interpretation as soon as it is processed (according to the results of Dell *et al.* 1983 and Nicol’s results summarized in Nicol & Swinney 1989) the *Frightener* role is immediately filled in. Thus after processing the words *It frightened*, the DECLARATIVE-S interpretation would fill in one of the argument roles of the *Frightening-Action* (as in (7.63), while the SUBJECT-EXTRAPOSITION interpretation would have filled in neither.

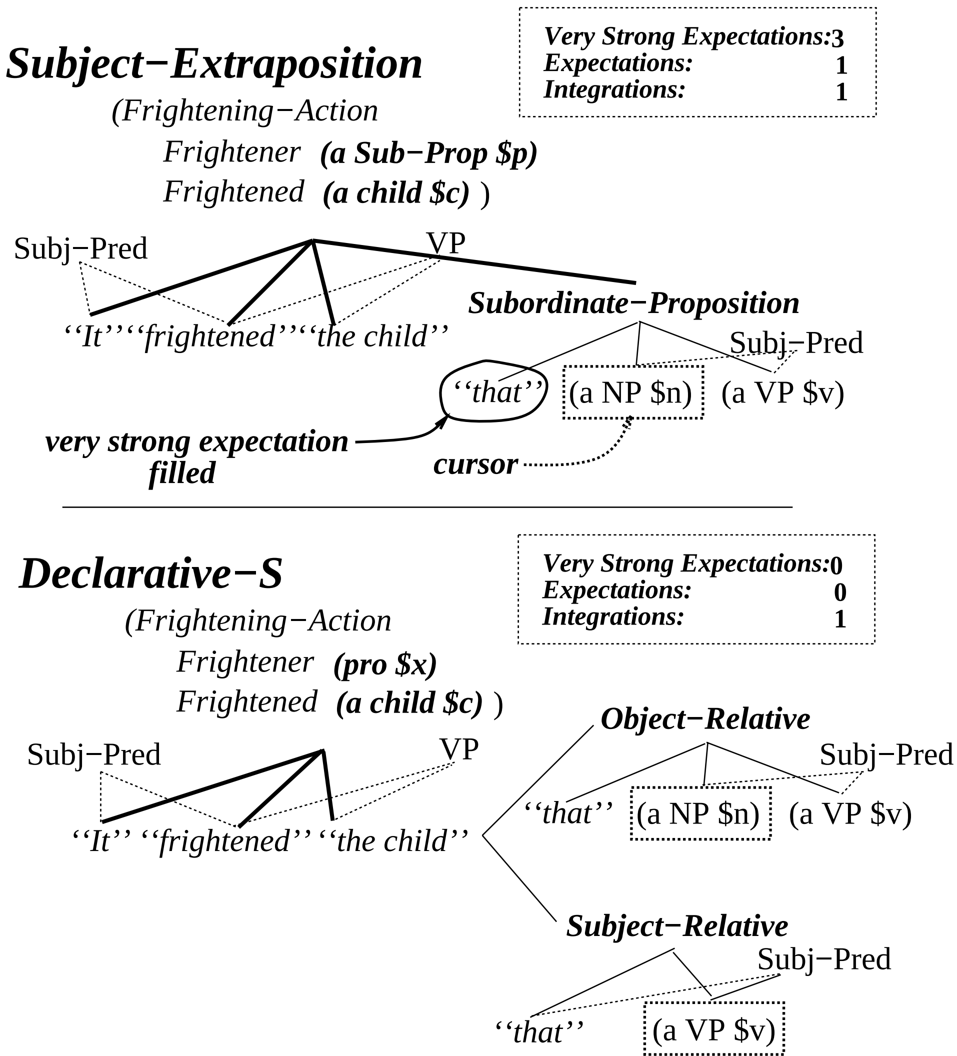


Figure 7.8: Processing an Extraposition (2): The Interpretation Store after *that*

(7.63) Frightening-Action $p
Frightener **(a explosion $w)**
Frightened

Chapter 8

Conclusions and Future Work

8.1 Conclusion

The grammar and the interpreter that this dissertation describes arose from an attempt to build a model which jointly incorporated the insights of artificial intelligence and natural language processing systems, of psycholinguistic models of processing, and of linguistic models of grammatical representation.

The model embodies a number of strong claims about sentence processing. One claim is *uniformity*. The interpreter is unified with respect to both representation and process. In the grammar, a single kind of knowledge structure, the *grammatical construction*, is used to represent lexical, syntactic, idiomatic, and semantic knowledge. CIG thus does not distinguish between the lexicon, the idiom dictionary, the syntactic rule base, and the semantic rule base. Uniformity in processing means that there is no distinction between the *lexical analyzer*, the *parser*, and the *semantic interpreter*. Because these kinds of knowledge are represented uniformly, they can be accessed, integrated, and disambiguated by a single mechanism, Sal.

A second claim the interpreter embodies is that sentence processing is fundamentally *knowledge-intensive* and expectation-based. Each aspect of Sal is knowledge-intensive; the representation theory, CIG, is based on representing every kind of linguistic knowledge. The access function is sensitive to top-down and bottom-up, syntactic and semantic knowledge, The integration function makes use of syntactic as well as semantic constraints on variables, and the selection function is based on coherence with grammatical knowledge as well as the semantic interpretation.

Attempting to express a coherent model of human sentence interpretation required limiting the domain of the model; the single-sentence interpreter ignores textual and intrasentential issues like reference and anaphora, as well as sub-lexical issues like phonology and orthography. Within the constraints of this limited domain, how successful has the model been in meeting the criteria expressed in Chapter 1?

The first criterion is *Functional Adequacy*. Functional adequacy required that an interpreter deal with a significantly large part of the interpretation problem. Chapter 1 claimed that a model which attempted to solve too small a problem might have problems scaling up to larger problems; a familiar problem in AI.

Although the model presented here is limited in scope to single-sentence interpretation, it does represent lexical, idiomatic, syntactic, and semantic rules, and it builds and disambiguates among interpretations that express high-level semantic structures. In particular, the model demonstrates that it is possible to build a system that can represent and integrate all of these kinds of information. In addition, the dissertation suggests in a few places how Sal might use the kind of information which is not currently modeled, such as the use of intra-sentential referential information to help disambiguate relative clauses or prepositional phrases.

The next criterion is *Representational Adequacy*. The Construction-Based Interpretive Grammar proposed in Chapter 3 represents linguistic knowledge at many levels, and accounts for traditional problems like the representation of long-distance dependencies and valence structures. The ability to describe weak as well as strong constructions allows a simple theoretical account of construction productivity and generalization.

The final criterion is *Psychological Adequacy*. The model is qualitatively consistent with a large body of psycholinguistic results, including:

- the on-line nature of the language interpretation process (see Chapter 6)
- the parallel nature and time course of lexical, idiomatic and syntactic access (see Chapter 5)
- the context-dependence of the access point (see Chapter 5)
- the use of frequency information in access and in selection (see Chapters 5 and 7)
- the use of lexical knowledge such as valence, subcategorization, and thematic roles in integration (see Chapter 6)
- the nature and time-course of gap-filling (see Chapter 6)
- the use of expectations in selection (see Chapter 7)

8.2 Problems and Future Work

The shortcomings of Sal and CIG which are, alas, all too numerous, can be grouped into two classes of limitations on the model; much of the natural future research on Sal and CIG consists of addressing these limitations.

The first shortcoming of the model is its size. The model is very small in scale in a number of ways, and needs to be expanded in all of them. The first size issue is the grammar. Both in terms of the extent of the theory, and the extent of the implementation, the current grammar, as described in Chapter 3, is far too small. The current implementation of CIG contains only about 50 constructions, and completely ignores pragmatic and morphological issues. A particularly necessary extension, and one I hope to pursue later, is a grammar of English noun-phrases, particularly the complex ordering constraints which occur in the noun-phrase. In a construction-based approach, such ordering constraints might be described as the results of semantic interactions. In addition, the grammar needs to be expanded to other languages. Extending the model to other languages, besides being necessary for the validity of the grammatical theory, will help point out deficiencies in the interpreter. Non-configurational languages as well as those with abundant ellipsis such as Japanese would be particularly useful in this regard.

A second size-related problem is the lack of frequency numbers for many constructions. The frequency numbers that are currently used were taken from previously published corpus statistics, such as Francis & Kučera (1982) for lexical and some larger constructions and Ellegård (1978) for other larger constructions. In many cases it was not possible to compute the frequency of a large construction from these sources. A better solution would be to do a more sophisticated analysis of a corpus directly — perhaps using a previously tagged corpus like the Penn Treebank Corpus.

Another size-related shortcoming has to do with the single-sentence nature of the model. By building on the solid foundation of the single-sentence interpreter, I hope to slowly add the capability to deal with multiple-sentence and textual input. A particularly important addition would be the use of discourse referent information to solve problems of anaphora and pronominal reference and to modify preferences for restrictive clause modification. Such knowledge has been used successfully in a number of interpreters, including those of Pereira & Pollack (1991), Hirst (1986), Winograd (1972) Mellish (1983), and Haddock (1989).

The second class of limitations on Sal concerns its symbolic nature; in effect, Sal is implemented with too coarse an algebra to account for the details of the time course of construction activation, or to account for the association effects shown by Reder (1983). This problem is also manifest in Sal's need for a distinct set of ranking criteria for the access and selection functions. Currently, the criteria which are used to access a construction include evidence from various top-down and bottom-up sources which are quite similar to the sorts of evidence used to rank interpretations for selections. Unifying these criteria, at least to the extent of allowing the access ranking metric to play a role in the selection ranking metric, would capture a significant generalization in the operation of the system.

However it is not possible for the access and selection models to be exactly equivalent, since psycholinguistic evidence indicates that access models must give greater weight to bottom-up factors, while successful selection models give greater weight to top-down factors. However, creating a uniform vocabulary for the description of these factors would improve the consistency of the theory.

In general, earlier models, such as Hirst (1986) or Cottrell (1985), have solved these problems by using connectionist or spreading-activation techniques, which include a fine enough algebra to account for detailed activation time-course and association effects, as well as allowing a unified metric for access and selection. We do not expect to be able to completely restructure the interpreter to incorporate a connectionist architecture. However, some parts of the model, particularly the access function and perhaps some parts of the selection function, seem quite amenable to connectionist implementation.

Bibliography

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

ABNEY, STEVEN P. 1991. Parsing by chunks. In *Principle-based parsing: Computation and psycholinguistics*, ed. by Robert C. Berwick, Steven P. Abney, & Carol Tenny, 257–278. Dordrecht: Kluwer.

ABNEY, STEVEN P. & MARK JOHNSON. 1991. Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20(3):233–250.

ADJUKIEWICZ, KAZIMIERZ. 1935/1967. Syntactic connexion. In *Polish logic*, ed. by Storrs McCall, 207–231. Oxford: Oxford University Press.

ADRIAENS, GEERT & STEVEN L. SMALL. 1988. Word Expert Parsing revisited in a cognitive science perspective. In *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, ed. by Steven L. Small, Garrison W. Cottrell, & Michael K. Tanenhaus. San Mateo, CA: Morgan Kaufmann.

AHO, ALBERT V., JOHN E. HOPCROFT, & JEFFERY D. ULLMAN. 1974. *The design and analysis of computer algorithms*. Reading, MA: Addison-Wesley.

AHO, ALBERT V., RAVI SETHI, & JEFFERY D. ULLMAN. 1986. *Compilers: Principles, techiques, and tools*. Reading, MA: Addison-Wesley.

AHO, ALBERT V. & JEFFERY D. ULLMAN. 1972. *The theory of parsing, translation, and compiling*, volume 1. Englewood Cliffs, NJ: Prentice-Hall.

ALLEN, JAMES. 1987. *Natural language understanding*. Menlo Park, CA: Benjamin Cummings.

ALTMANN, GERRY. 1988. Ambiguity, parsing strategies, and computational models. *Language and Cognitive Processes*, 3(2):73–97.

ALTMANN, GERRY T. M. & MARK J. STEEDMAN. 1988. Interaction with context during human sentence processing. *Cognition*, 30:191–238.

ARONOFF, MARK. 1976. *Word-formation in generative grammar*. Linguistic Inquiry Monograph no. 1. Cambridge, MA: MIT Press.

BACH, EMMON. 1976. An extension of classical transformational grammar. In *Problems of Linguistic Metatheory (Proceedings of the 1976 Conference)*. Michigan State University.

BAHL, LALIT R., FREDERICK JELINEK, & ROBERT L. MERCER. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190.

BAKER, JAMES K. 1975/1990. Stochastic modeling for automatic speech understanding. In *Readings in speech recognition*, ed. by Alex Waibel & Kai-Fu Lee, 297–307. Los Altos: Morgan Kaufmann. Originally appeared in *Speech Recognition*, Academic Press, 1975.

BARD, ELLEN GURMAN. 1990. Competition, lateral inhibition, and frequency. In *Cognitive models of speech processing*, ed. by Gerry T. M. Altmann, 185–210. Cambridge, MA: MIT Press.

BARTON, JR, G. EDWARD, ROBERT C. BERWICK, & ERIC SVEN RISTAD. 1987. *Computational complexity and natural language*. Cambridge, MA: MIT Press.

BAZELL, C. E. 1952/1966. The correspondence fallacy in structural linguistics. In *Studies by members of the English department, Istanbul University (3), reprinted in Readings in linguistics II (1966)*, ed. by Eric P. Hamp, Fred W. Householder, & Robert Austerlitz, 271–298. Chicago: University of Chicago Press.

BECKER, CURTIS A. 1980. Semantic context effects in visual word recognition: An analysis of semantic strategies. *Memory & Cognition*, 8(6):493–512.

BECKER, JOSEPH. 1975. The phrasal lexicon. In *Proceedings of TINLAP-1 (Theoretical Issues in Natural Language Processing)*, ed. by Roger C. Schank & Bonnie L. Nash-Webber, 60–63, Cambridge, MA.

BERWICK, ROBERT C. 1991. Principles of principle-based parsing. In *Principle-based parsing: Computation and psycholinguistics*, ed. by Robert C. Berwick, Steven P. Abney, & Carol Tenny, 1–38. Dordrecht: Kluwer.

BERWICK, ROBERT C. & AMY S. WEINBERG. 1983. The role of grammars in models of language use. *Cognition*, 13:1–61.

BERWICK, ROBERT C. & AMY S. WEINBERG. 1984. *The grammatical basis of linguistic performance*. Cambridge, MA: MIT Press.

BEVER, THOMAS G. 1970. The cognitive basis for linguistic structures. In *Cognition and the development of language*, ed. by John R. Hayes. New York: Wiley.

BIRNBAUM, LAWRENCE & MALLORY SELFRIDGE. 1981. Conceptual analysis of natural language. In *Inside computer understanding: Five programs plus miniatures*, ed. by Roger C. Schank & Christopher K. Riesbeck, 318–353. Hillsdale: Erlbaum.

BLOOMFIELD, LEONARD. 1933. *Language*. Chicago: University of Chicago Press.

BOAS, HANS ULRICH. 1975. *Syntactic generalizations and linear order in generative transformational grammar*. Tübingen: Verlag Gunter Narr.

BOBROW, DANIEL & TERRY WINOGRAD. 1977. An overview of KRL: A knowledge representation language. *Cognitive Science*, 1:346.

BOBROW, ROBERT J. & BONNIE WEBBER. 1980. Knowledge representation for syntactic/semantic processing. In *Proceedings of the First National Conference on Artificial Intelligence*, 316–323. Morgan Kaufmann.

BOLAND, JULIE E., 1991. *The use of lexical knowledge in sentence processing*. Rochester: University of Rochester dissertation.

BOLAND, JULIE E. & MICHAEL K. TANENHAUS. 1991. The role of lexical representations in sentence processing. In *Understanding word and sentence*, ed. by G. B. Simpson, 331–366. Elsevier Science.

BOLAND, JULIE E., MICHAEL K. TANENHAUS, GREG CARLSON, & SUSAN M. GARNSEY. 1989. Lexical projection and the interaction of syntax and semantics in parsing. *Journal of Psycholinguistic Research*, 18(6):563–576.

BOLAND, JULIE E., MICHAEL K. TANENHAUS, & SUSAN M. GARNSEY. 1990. Evidence for the immediate use of verb control information in sentence processing. *Journal of Memory and Language*, 29:413–432.

BOLINGER, DWIGHT. 1971. *The phrasal verb in English*. Cambridge, MA: Harvard University Press.

BOLINGER, DWIGHT. 1979. Meaning and memory. In *Experience forms: Their cultural and individual place and function*, ed. by George G. Haydu, 95–111. The Hague: Mouton.

BRACHMAN, RONALD J. & JAMES G. SCHMOLZE. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216.

BRESNAN, JOAN. 1978. A realistic transformational grammar. In *Linguistic theory and psychological reality*, ed. by Morris Halle, Joan Bresnan, & George A. Miller, 1–59. Cambridge, MA: MIT Press.

BRESNAN, JOAN (ed.). 1982a. *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.

BRESNAN, JOAN. 1982b. Polyadicity. In *The mental representation of grammatical relations*, ed. by Joan Bresnan, 149–172. Cambridge, MA: MIT Press.

BRESNAN, JOAN & RONALD M. KAPLAN. 1982. Introduction: Grammars as mental representations of language. In *The mental representation of grammatical relations*, ed. by Joan Bresnan. Cambridge, MA: MIT Press.

BRUGMAN, CLAUDIA & GEORGE LAKOFF. 1988. Cognitive topology and lexical networks. In *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, ed. by Steven L. Small, Garrison W. Cottrell, & Michael K. Tanenhaus. San Mateo: Morgan Kaufmann.

BUTTERWORTH, BRIAN. 1983. Lexical representation. In *Language production, volume 2: Development, writing and other language processes*, ed. by Brian Butterworth, 257–294. New York: Academic.

CACCIARI, CRISTINA & PATRIZIA TABOSSI. 1988. The comprehension of idioms. *Journal of Memory and Language*, 27:668–683.

CARDIE, CLAIRE & WENDY LEHNERT. 1991. A cognitively plausible approach to understanding complex syntax. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 117–124. Morgan Kaufmann.

CARLSON, GREG N. & MICHAEL K. TANENHAUS. 1987. Thematic roles and language comprehension. In *Thematic relations*, ed. by Wendy Wilkins, 263–288. San Diego: Academic Press.

CATER, ARTHUR. 1983. Request-based parsing with low-level syntactic recognition. In *Automatic natural language parsing*, ed. by Karen Sparck Jones & Yorick Wilks, 141–147. Chichester: Ellis-Horwood.

CHARNIAK, EUGENE. 1981. The case-slot identity theory. *Cognitive Science*, 5(3):285–292.

CHARNIAK, EUGENE. 1983. A parser with something for everyone. In *Parsing natural language*, ed. by Margaret King, 117–149. London: Academic Press.

CHARNIAK, EUGENE & ROBERT GOLDMAN. 1988. A logic for semantic interpretation. In *Proceedings of the 26th Annual Conference of the Association for Computational Linguistics*, Buffalo, NY.

CHOMSKY, NOAM. 1956/1975. *The logical structure of linguistic theory*. Plenum.

CHOMSKY, NOAM. 1957. *Syntactic structures*. The Hague: Mouton.

CHOMSKY, NOAM. 1962. A transformational approach to syntax. In *The structure of language, readings in the philosophy of language*, ed. by Jerry A. Fodor & Jerrold J. Katz, 211–245. Englewood Cliffs, NJ: Prentice Hall.

CHOMSKY, NOAM. 1964. *Current issues in linguistic theory*. Number XXXVIII in Janua Linguarum Series Minor. The Hague: Mouton and Co.

CHOMSKY, NOAM. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

CHOMSKY, NOAM. 1966. *Cartesian linguistics: A chapter in the history of rationalist thought*. New York: Harper & Row.

CHOMSKY, NOAM. 1970. Remarks on nominalization. In *Readings in English transformational grammar*, ed. by R. Jacobs & P. Rosenbaum, 184–221. Waltham, MA: Ginn.

CHOMSKY, NOAM. 1972. *Language and mind*. New York: Harcourt Brace Jovanovich.

CHOMSKY, NOAM. 1981. *Lectures on government and binding*. Dordrecht: Foris.

CHOMSKY, NOAM. 1986. *Knowledge of language: Its nature, origin, and use*. New York: Praeger.

CHURCH, KENNETH & RAMESH PATIL. 1982. Coping with syntactic ambiguity. *American Journal of Computational Linguistics*, 8(3-4):139–149.

CLIFTON, JR, CHARLES & FERNANDA FERREIRA. 1987. Modularity in sentence comprehension. In *Modularity in knowledge representation and natural-language understanding.*, ed. by Jay L. Garfield, 277–290. Cambridge, MA: MIT Press.

CLIFTON, JR, CHARLES & FERNANDA FERREIRA. 1989. Ambiguity in context. *Language and Cognitive Processes*, 4:77–103.

CLIFTON, JR, CHARLES & LYN FRAZIER. 1989. Comprehending sentences with long-distance dependencies. In *Linguistic structure in language processing*, ed. by G. N. Carlson & M. K. Tanenhaus, 273–317. Dordrecht: Kluwer Academic.

CLIFTON, JR, CHARLES, LYN FRAZIER, & CYNTHIA CONNINE. 1984. Lexical expectations in sentence comprehension. *Journal of Verbal Learning and Verbal Behavior*, 23:696–708.

COLLINS, A. M. & M. ROSS QUILLIAN. 1969. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8.

CORREA, NELSON. 1991. Empty categories, chain binding, and parsing. In *Principle-based parsing: Computation and psycholinguistics*, ed. by Robert C. Berwick, Steven P. Abney, & Carol Tenny, 83–122. Dordrecht: Kluwer.

COTTER, C. A. 1984. Inferring indirect objects in sentences: Some implications for the semantics of verbs. *Language and speech*, 27:25–45.

COTTRELL, GARRISON W. 1985. Connectionist parsing. In *Program of the Seventh Annual Conference of the Cognitive Science Society*, 201–211, Irvine, CA.

COTTRELL, GARRISON W. 1989. *A connectionist approach to word sense disambiguation*. San Mateo: Morgan Kaufmann.

CRAIN, STEPHEN & JANET DEAN FODOR. 1985. How can grammars help parsers? In *Natural language parsing*, ed. by David R. Dowty, Lauri Kartunnen, & Arnold Zwicky, 94–128. Cambridge: Cambridge University Press.

CRAIN, STEPHEN & MARK STEEDMAN. 1985. On not being led up the garden path: the use of context by the psychological syntax processor. In *Natural language parsing*, ed. by David R. Dowty, Lauri Kartunnen, & Arnold Zwicky, 320–358. Cambridge: Cambridge University Press.

CULLINGFORD, RICHARD E. 1981. SAM. In *Inside computer understanding: Five programs plus miniatures*, ed. by Roger C. Schank & Christopher K. Riesbeck, 75–119. Hillsdale: Erlbaum.

CUTLER, ANNE. 1983. Lexical complexity and sentence processing. In *The process of language*, ed. by G. B. Flores d'Arcais & R. J. Jarvella, 43–79. New York: Wiley.

DAHLGREN, KATHLEEN & J. MCDOWELL. 1986. Using commonsense knowledge to disambiguate prepositional phrase modifiers. *Proceedings of the Fifth National Conference on Artificial Intelligence*, 1:589–593.

DEJONG, GERALD. 1982. An overview of the FRUMP system. In *Strategies for natural language processing*, ed. by Wendy G. Lehnert & Martin H. Ringle, 149–176. New Jersey: Lawrence Erlbaum.

DELL, GARY S., GAIL MCKOON, & ROGER RATCLIFF. 1983. The activation of antecedent information during the processing of anaphoric reference in reading. *Journal of Verbal Learning and Verbal Behavior*, 22:121–132.

DERWING, BRUCE L. 1973. *Transformational grammar as a theory of language acquisition*. Cambridge: Cambridge University Press.

DERWING, BRUCE L. & ROYAL SKOUSEN. 1989. Real-time morphology. In *Proceedings of the 15th Annual Meeting of the Berkeley Linguistics Society*, 48–62, Berkeley, CA.

EARLEY, JAY. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 6(8):451–455.

ELLEGÅRD, ALVAR. 1978. *The syntactic structure of English texts: A computer-based study of four kinds of text in the Brown University Corpus*. Number 43 in Gothenburg Studies in English. Göteborg: Acta Universitatis Gothoburgensis.

ERMAN, LEE D., FREDERICK HAYES-ROTH, VICTOR R. LESSER, & D. RAJ REDDY. 1980/1981. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. In *Readings in artificial intelligence*, ed. by Bonnie Lynn Webber & Nils J. Nilsson, 349–389. Los Altos: Morgan Kaufmann. Originally appeared in *Computing Surveys* 12(2).

FAHLMAN, SCOTT E. 1979. *NETL: A system for representing and using real-world knowledge*. Cambridge, MA: MIT Press.

FENSTAD, JENS ERIK, PER-KRISTIAN HALVORSEN, TORE LANGHOLM, & JOHAN VAN BENTHEM. 1985. Equations, schemata, and situations: A framework for linguistic semantics. Technical Report CSLI-85-29, CSLI, Stanford.

FILLMORE, CHARLES J. 1968. The case for case. In *Universals in linguistic theory*, ed. by Emmon W. Bach & Robert T. Harms, 1–88. New York: Holt, Rinehart & Winston.

FILLMORE, CHARLES J. 1977. The case for case reopened. In *Syntax and semantics volume 8: Grammatical relations*, ed. by Peter Cole & Jerrold M. Sadock. New York: Academic Press.

FILLMORE, CHARLES J. 1978. On the organization of semantic information in the lexicon. In *Papers from the Parasession on the Lexicon*, ed. by Donka Farkas, Wesley M. Jacobsen, & Karol W. Todrys, 148–173, Chicago. Chicago Linguistic Society.

FILLMORE, CHARLES J., 1985. Some worries about phrase structure. Lecture notes, Linguistics Group Meeting, September 10, 1985.

FILLMORE, CHARLES J. 1986. Pragmatically controlled zero anaphora. In *Proceedings of the 12th Annual Meeting of the Berkeley Linguistics Society*, 95–107, Berkeley, CA.

FILLMORE, CHARLES J. 1989a. The mechanisms of “construction grammar”. In *Proceedings of the 15th Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA.

FILLMORE, CHARLES J., 1989b. On grammatical constructions. Course notes draft.

FILLMORE, CHARLES J., PAUL KAY, & M. C. O’CONNOR. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64(3):501–538.

FISH, STANLEY. 1970. Literature in the reader: Affective stylistics. *New Literary History*, II(1):123–162.

FLICKINGER, DANIEL, CARL POLLARD, & THOMAS WASOW. 1985. Structure-sharing in lexical representation. In *Proceedings of the 23rd Annual Conference of the Association for Computational Linguistics*, 262–267, Chicago.

FODOR, JANET DEAN. 1978. Parsing strategies and constraints on transformations. *Linguistic Inquiry*, 9(3):427–473.

FODOR, JANET DEAN. 1989. Empty categories in sentence processing. *Language and Cognitive Processes*, 4(3/4):155–209.

FODOR, JERRY, THOMAS BEVER, & MERRILL GARRETT. 1974. *The psychology of language*. New York: McGraw-Hill.

FODOR, JERRY A. 1983. *Modularity of mind*. Cambridge, MA: MIT Press.

FODOR, JERRY A. & JERROLD J. KATZ. 1964. *The structure of language, readings in the philosophy of language*. Englewood Cliffs, NJ: Prentice Hall.

FOLEY, WILLIAM A. & ROBERT D. VAN VALIN, JR. 1984. *Functional syntax and universal grammar*. Cambridge: Cambridge University Press.

FONG, SANDIWAY. 1991. The computational implementation of principle-based parsers. In *Principle-based parsing: Computation and psycholinguistics*, ed. by Robert C. Berwick, Steven P. Abney, & Carol Tenny, 65–82. Dordrecht: Kluwer.

FORD, MARILYN, JOAN BRESNAN, & RONALD M. KAPLAN. 1982. A competence-based theory of syntactic closure. In *The mental representation of grammatical relations*, ed. by Joan Bresnan. Cambridge, MA: MIT Press.

FRANCIS, W. NELSON & HENRY KUČERA. 1982. *Frequency analysis of English usage*. Boston: Houghton Mifflin.

FRASER, BRUCE. 1976. *The verb-particle combination in English*. New York: Academic.

FRAZIER, LYN, 1978. *On comprehending sentences: syntactic parsing strategies*. Connecticut: University of Connecticut dissertation. Dissertation distributed by the Indiana University Linguistics Club.

FRAZIER, LYN. 1987a. Sentence processing: A tutorial review. In *Attention and performance XII*, ed. by M. Coltheart, 559–586. Hillsdale, NJ: Erlbaum.

FRAZIER, LYN. 1987b. Theories of sentence processing. In *Modularity in knowledge representation and natural-language understanding*, ed. by Jay L. Garfield, 291–307. Cambridge, MA: MIT Press.

FRAZIER, LYN & JANET DEAN FODOR. 1978. The sausage machine: A new two-stage parsing model. *Cognition*, 6:291–295.

FU, K. S. 1974. *Syntactic methods in pattern recognition*, volume 112 of *Mathematics in science and engineering*. New York: Academic.

FUJISAKI, T., F. JELINEK, J. COCKE, & E. BLACK. 1991. A probabilistic parsing method for sentence disambiguation. In *Current issues in parsing technology*, ed. by Masaru Tomita, 139–152. Boston: Kluwer.

FUJISAKI, TETSUNOSUKE. 1984. A stochastic approach to sentence parsing. In *Proceedings of the Tenth International Conference on Computational Linguistics*, 16–19, Stanford, California. Stanford University.

GARNSEY, SUSAN M., MICHAEL K. TANENHAUS, & ROBERT M. CHAPMAN. 1989. Evoked potentials and the study of sentence comprehension. *Journal of Psycholinguistic Research*, 18(1):51–60.

GARROD, SIMON & ANTHONY SANFORD. 1981. Bridging inferences and the extended domain of reference. In *Attention and performance IX*, ed. by John Long & Alan Baddeley, 331–346. Hillsdale, NJ: Erlbaum.

GARROD, SIMON & ANTHONY SANFORD. 1990. Referential processes in reading: Focusing on roles and individuals. In *Comprehension processes in reading*, ed. by D.A. Balota, G. B. Flores d'Arcais, & K. Rayner, 465–485. New Jersey: Lawrence Erlbaum.

GARROD, SIMON & ANTHONY SANFORD. 1991. On the real-time character of interpretation during reading. *Language and Cognitive Processes*, 1:43–59.

GAWRON, JEAN MARK, 1983. *Lexical representations and the semantics of complementation*. Berkeley, CA: University of California dissertation.

GAZDAR, GERALD. 1981. Unbounded dependencies and coordinate structure. *Linguistic Inquiry*, 12(2):155–184.

GAZDAR, GERALD. 1982. Phrase structure grammar. In *The nature of syntactic representation*, ed. by Pauline Jacobson & Geoffrey K. Pullum, 131–186. Dordrecht: Reidel.

GAZDAR, GERALD, EWAN KLEIN, GEOFFREY PULLUM, & IVAN SAG. 1985. *Generalized phrase structure grammar*. Oxford: Basil Blackwell.

GERSHMAN, ANATOLE V. 1982. A framework for conceptual analyzers. In *Strategies for natural language processing*, ed. by Wendy G. Lehnert & Martin H. Ringle, 177–201. Hillsdale: Erlbaum.

GIBBS, JR, RAYMOND W., NANDINI P. NAYAK, & COOPER CUTTING. 1989. How to kick the bucket and not decompose: Analyzability and idiom processing. *Journal of Memory and Language*, 28:576–593.

GIBSON, EDWARD. 1990a. Memory capacity and sentence processing. In *Proceedings of the 28th Annual Conference of the Association for Computational Linguistics*.

GIBSON, EDWARD. 1990b. Recency preference and garden-path effects. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, 372–379, Cambridge, MA.

GIBSON, EDWARD, 1991. *A computational theory of human linguistic processing: Memory limitations and processing breakdown*. Pittsburgh, PA: Carnegie Mellon University dissertation.

GLUCKSBERG, S., R. J. KREUZ, & S. H. RHO. 1986. Context can constrain lexical access: Implications for modles of language comprehension. *Journal of Experimental Psychology: Learning Memory, and Cognition*, 12:323–335.

GOLDBERG, ADELE E. 1989. A unified account of the English ditransitive. In *Proceedings of the 15th Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA.

GOLDBERG, ADELE E. 1991. On the problems with lexical rule accounts of argument structure. In *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, 729–733, Chicago.

GORRELL, PAUL G., 1987. *Studies of human syntactic processing: Ranked-parallel versus serial models*. Connecticut: University of Connecticut dissertation.

GORRELL, PAUL G. 1989. Establishing the loci of serial and parallel effects in syntactic processing. *Journal of Psycholinguistic Research*, 18(1):61–73.

GRIFFITHS, T. V. & S. R. PETRICK. 1965. On the relative efficiencies of context-free grammar recognizers. *Communications of the ACM*, 8(5):289–300.

GRIMSHAW, JANE. 1979. Complement selection and the lexicon. *Linguistic Inquiry*, 10(2):279–326.

GROSS, MAURICE. 1984. Lexicon-grammar and the syntactic analysis of French. *Proceedings of the Tenth International Conference on Computational Linguistics*, 275–282.

HADDOCK, NICHOLAS J. 1989. Computational models of incremental semantic interpretation. *Language and Cognitive Processes*, 4:337–368.

HALE, KEN. 1983. Warlpiri and the grammar of non-configurational languages. *Natural Language & Linguistic Theory*, 1(3):5–47.

HALLE, MORRIS. 1959. *The sound pattern of Russian; A linguistic and acoustical investigation*. The Hague: Mouton.

HALLIDAY, M. A. K. 1985. *An introduction to functional grammar*. London: Edward Arnold.

HANKAMER, JORGE. 1973. Unacceptable ambiguity. *Linguistic Inquiry*, IV(1):17–68.

HANKAMER, JORGE. 1989. Morphological parsing and the lexicon. In *Lexical representation and process*, ed. by William Marslen-Wilson, 392–408. Cambridge, MA: MIT Press.

HARRIS, ZELLIG S. 1946. From morpheme to utterance. *Language*, 22(3):161–183.

HAUSSER, ROLAND R. 1986. *NEWCAT: parsing natural language using left-associative grammar*. Berlin: Springer-Verlag.

HAYS, DAVID G. 1966. Parsing. In *Readings in automatic language processing*, ed. by David G. Hays, 73–82. New York: American Elsevier.

HENDRIX, GARY. 1978/1986. Semantic aspects of translation. In *Readings in natural language processing*, ed. by Barbara J. Grosz, Karen Sparck-Jones, & Bonnie Lynn Webber, 267–283. Los Altos: Morgan Kaufmann. Originally appeared in Don Walker (ed.), *Understanding spoken language*, 1978.

HILL, ARCHIBALD A. 1970. Laymen, lexicographers, and linguists. *Language*, 46(2):245–258.

HIRST, GRAEME. 1981. *Anaphora in natural language understanding: A survey*. Number 119 in Lecture notes in computer science. New York: Springer-Verlag.

HIRST, GRAEME. 1984. A semantic process for syntactic disambiguation. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, 148–152.

HIRST, GRAEME. 1986. *Semantic interpretation and the resolution of ambiguity*. Cambridge: Cambridge University Press.

HIRST, GRAEME & EUGENE CHARNIAK. 1982. Word sense and case slot disambiguation. In *Proceedings of the Second National Conference on Artificial Intelligence*, 95–98.

HOBBS, JERRY R. & JOHN BEAR. 1990. Two principles of parse preference. In *Proceedings of the 13th International Conference on Computational Linguistics*, 162–167, Helsinki.

HOBBS, JERRY R., MARK STICKEL, PAUL MARTIN, & DOUGLAS EDWARDS. 1988. Interpretation as abduction. In *Proceedings of the 26th Annual Conference of the Association for Computational Linguistics*, 95–103, Buffalo, NY.

HOCKETT, CHARLES F. 1954. Two models of grammatical description. *Word*, 10:210–231.

HOGABOAM, T. W. & C. A. PERFETTI. 1975. Lexical ambiguity and sentence comprehension. *Journal of Verbal Learning and Verbal Behavior*, 14:265–274.

HOLBROOK, JENNIFER K., KURT P. EISELT, RICHARD H. GRANGER, JR, & EDWARD H. MATTHEI. 1988. (almost) never letting go: Inference retention during text understanding. In *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, ed. by Steven L. Small, Garrison W. Cottrell, & Michael K. Tanenhaus. San Mateo, CA: Morgan Kaufmann.

HUCK, GEOFFREY J. & ALMERINDO E. OJEDA. 1987. Introduction. In *Syntax and semantics 20: Discontinuous constituency*, ed. by Geoffrey J. Huck & Almerindo E. Ojeda, 1–10. Academic Press.

HUDSON, RICHARD. 1984. *Word grammar*. Oxford: Basil Blackwell.

HUMBOLDT, WILHELM VON. 1836/1988. *On language: The diversity of human language structure and its influence on the mental development of mankind*. Cambridge: Cambridge University Press. Translated by Peter Heath. Originally published as the introduction to the author's *Über die Kavi-Sprache auf der Insel Java (1836–1840)* and also issued seperately with the title *Über die Verschiedenheit des Menschlichen Sprachbaues und ihren Einfluss auf die geistige Entwickelung des Menchengeschlechts*.

HYMES, DELL & JOHN FOUGHT. 1981. *American structuralism*. Janua Linguarum Series Minor, no. 102. The Hague: Mouton.

INGRIA, ROBERT J. P. 1990. The limits of unification. In *Proceedings of the 28th Annual Conference of the Association for Computational Linguistics*, 194–204, Pittsburgh, PA.

JACKENDOFF, RAY. 1975. Morphological and semantic regularities in the lexicon. *Language*, 51(3):639–671.

JACKENDOFF, RAY. 1977. *X-bar syntax: A study of phrase structure*. Linguistic Inquiry Monograph no. 2. Cambridge, MA: MIT Press.

JACOBS, PAUL, 1985. *A knowledge-based approach to language generation*. Berkeley, CA: University of California dissertation. Available as University of California at Berkeley Computer Science Division Technical Report #86/254.

JAIN, AJAY N. & ALEX H. WAIBEL. 1991. Parsing with connectionist networks. In *Current issues in parsing technology*, ed. by Masaru Tomita, 243–260. Boston: Kluwer.

JELINEK, FREDERICK & JOHN D. LAFFERTY. 1991. Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics*, 17(3):315–323.

JOHNSON, MARK. 1991. Deductive parsing: The use of knowledge of language. In *Principle-based parsing: Computation and psycholinguistics*, ed. by Robert C. Berwick, Steven P. Abney, & Carol Tenny, 39–64. Dordrecht: Kluwer.

JOOS, MARTIN. 1958. Review of Eric Hamp *A glossary of American technical linguistic usage* . *Language*, 34:279–288.

JOOS, MARTIN. 1972. Semantic axiom number one. *Language*, 48(2):257–265.

JOSHI, ARAVIND K. 1985. Tree adjoining grammars: how much context-sensitivity is required to provide reasonable structural descriptions? In *Natural language parsing*, ed. by David R. Dowty, Lauri Kartunnen, & Arnold Zwicky, 206–250. Cambridge: Cambridge University Press.

JOSHI, ARAVIND K. & K. VIJAY-SHANKER. 1989. Treatment of long distance dependencies in LFG and TAG: Functional uncertainty in LFG is a corollary in TAG. In *Proceedings of the 27th Annual Conference of the Association for Computational Linguistics*, 220–227, Vancouver, Canada.

JURAFSKY, DANIEL. 1988. Issues in relating syntax and semantics. In *Proceedings of the 12th International Conference on Computational Linguistics*, 278–284, Budapest.

JURAFSKY, DANIEL. 1990. Representing and integrating linguistic knowledge. In *Proceedings of the 13th International Conference on Computational Linguistics*, 199–204, Helsinki.

JURAFSKY, DANIEL. 1991. An on-line model of human sentence interpretation. In *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, 449–454, Chicago.

JUST, MARCEL ADAM & PATRICIA A. CARPENTER. 1980. A theory of reading: From eye fixations to comprehension. *Psychological Review*, 87(4):123–154.

KAPLAN, RONALD M. 1972. Augmented Transition Networks as psychological models of sentence comprehension. *Artificial Intelligence*, 3:77–100.

KAPLAN, RONALD M. 1973. A general syntactic processor. In *Natural language processing*, ed. by Randall Rustin, 193–241. New York: Algorithmics Press.

KAPLAN, RONALD M. & JOHN T. MAXWELL, III. 1988. An algorithm for functional uncertainty. In *Proceedings of the 12th International Conference on Computational Linguistics*, 297–302, Budapest.

KAPLAN, RONALD M. & ANNIE ZAENEN. 1989. Long-distance dependencies, constituent structure, and functional uncertainty. In *Alternative conceptions of phrase structure*, ed. by Mark R. Baltin & Anthony S. Kroch, 17–42. Chicago: University of Chicago.

KARLGREN, HANS. 1976. Why trees in syntax. *Studia Linguistica*, 30(1):1–33.

KARTTUNEN, LAURI. 1989. Radical lexicalism. In *Alternative conceptions of phrase structure*, ed. by Mark R. Baltin & Anthony S. Kroch, 43–65. Chicago: University of Chicago.

KAY, MARTIN. 1973. The MIND system. In *Natural language processing*, ed. by Randall Rustin, 155–188. New York: Algorithmics Press.

KAY, MARTIN. 1979. Functional grammar. In *Proceedings of the 5th Annual Meeting of the Berkeley Linguistics Society*, 142–158, Berkeley, CA.

KAY, MARTIN. 1982. Algorithm schemata and data structures in syntactic processing. In *Text processing: Text analysis and generation, text typology and attribution*, ed. by Sture Allén, 327–358. Stockholm: Almqvist and Wiksell.

KAY, PAUL. 1990. Even. *Linguistics and Philosophy*, 13:59–216.

KIMBALL, JOHN P. 1973. Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47.

KIMBALL, JOHN P. 1975. Predictive analysis and over-the-top parsing. In *Syntax and semantics 4*, ed. by John P. Kimball, 155–179. Academic Press.

KINTSCH, WALTER. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95(2):163–182.

KUNO, SUSUMU. 1965. The predictive analyzer and a path elimination technique. *Communications of the ACM*, 8(7):453–462.

KUNO, SUSUMU & A.G. OETTINGER. 1962/1986. Multiple path syntactic analyzer. In *Readings in natural language processing*, ed. by Barbara J. Grosz, Karen Sparck-Jones, & Bonnie Lynn Webber, 17–23. Los Altos: Morgan Kaufmann. 1986.

KURTZMAN, HOWARD S., 1985. *Studies in syntactic ambiguity resolution.* Cambridge, MA: MIT dissertation.

KURTZMAN, HOWARD S., LOREN F. CRAWFORD, & CAYLEE NYCHIS-FLORENCE. 1991. Locating wh- traces. In *Principle-based parsing: Computation and psycholinguistics*, ed. by Robert C. Berwick, Steven P. Abney, & Carol Tenny, 347–382. Dordrecht: Kluwer.

LAKOFF, GEORGE. 1977. Linguistic gestalts. In *Papers from the Thirteenth Regional Meeting of the Chicago Linguistic Society*, ed. by Woodford A. Beach, Samuel E. Fox, & Shulamith Philosoph, 236–287, Chicago.

LAKOFF, GEORGE. 1984. There-constructions: A case study in grammatical construction theory and prototype theory. Technical Report 18, University of California at Berkeley Cognitive Science, Berkeley, CA.

LAKOFF, GEORGE. 1987. *Women, fire, and dangerous things*. Chicago: University of Chicago Press.

LANGACKER, RONALD. 1987. *Foundations of cognitive grammar I: Theoretical prerequisites*. Stanford: Stanford University Press.

LANGENDOEN, D. TERENCE & PAUL M. POSTAL. 1984. *The vastness of natural languages*. Oxford: Basil Blackwell.

LYTINEN, STEVEN L. 1986. Dynamically combining syntax and semantics in natural language processing. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, volume 1, 574–578. Morgan Kaufmann.

LYTINEN, STEVEN L. 1991. Semantics-first natural language processing. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 111–116. Morgan Kaufmann.

MACDONALD, MARYELLEN C., MARCEL ADAM JUST, & PATRICIA A. CARPENTER. in press. Working memory constraints on the processing of syntactic ambiguity. *Cognitive Psychology*.

MAIDA, ANTHONY S. & STUART C. SHAPIRO. 1982. Intensional concepts in propositional semantic networks. *Cognitive Science*, 6(4):291–330.

MAKKAI, ADAM. 1972. *Idiom structure in English*. The Hague: Mouton.

MARCUS, MITCHELL P. 1980. *A theory of syntactic recognition for natural language*. Cambridge, MA: MIT Press.

MARCUS, MITCHELL P., DONALD HINDLE, & MARGARET M. FLECK. 1983. D-theory: Talking about talking about trees. In *Proceedings of the 21st Annual Conference of the Association for Computational Linguistics*, 129–136, Cambridge, MA.

MARSLEN-WILSON, WILLIAM. 1975. Sentence perception as an interactive parallel process. *Science*, 189:226–228.

MARSLEN-WILSON, WILLIAM. 1987. Functional parallelism in spoken word recognition. *Cognition*, 25:71–102.

MARSLEN-WILSON, WILLIAM. 1990. Activation, competition, and frequency in lexical access. In *Cognitive models of speech processing*, ed. by Gerry T. M. Altmann, 148–172. Cambridge, MA: MIT Press.

MARSLEN-WILSON, WILLIAM, COLIN M. BROWN, & LORRAINE KOMISARJEVSKY TYLER. 1988. Lexical representations in spoken language comprehension. *Language and Cognitive Processes*, 3(1):1–16.

MARTIN, JAMES H. 1990. *A computational model of metaphor interpretation*. Academic Press.

MCCAWLEY, JAMES D. 1968. Concerning the base component of a transformational grammar. *Foundations of Language*, 4(3):243–269.

MCCLELLAND, JAMES L. 1987. The case for interactionism in language processing. In *Attention and performance XII*, ed. by M. Coltheart, 3–36. Hillsdale, NJ: Erlbaum.

MCCLELLAND, JAMES L. & JEFFREY L. ELMAN. 1986. Interactive processes in speech perception: The TRACE model. In *Parallel distributed processing volume 2: Psychological and biological models*, ed. by James L. McClelland, David E. Rumelhart, & the PDP Research Group, 58–121. Cambridge, MA: MIT Press.

MCCLELLAND, JAMES L., MARK ST. JOHN, & ROMAN TARABAN. 1989. Sentence comprehension: A parallel distributed processing approach. *Language and Cognitive Processes*, 4(3/4):123–154.

MCCORD, MICHAEL C. 1982. Using slots and modifiers in logic grammars for natural language. *Artificial Intelligence*, 18(3):327–368.

MCCORD, MICHAEL C. 1990. Slot grammar: A system for simpler construction of practical natural language grammars. In *Natural language and logic*, ed. by R. Studer, 118–145. Berlin: Springer-Verlag.

MCKOON, GAIL & ROGER RATCLIFF. 1990. Dimensions of inference. In *Inferences and text comprehension*, ed. by Arthur C. Graesser & Gordon H. Bower, 313–328. Academic Press.

MCROY, SUSAN WEBER & GRAEME HIRST. 1990. Race-based parsing and syntactic disambiguation. *Cognitive Science*, 14:313–353.

MELLISH, CHRISTOPHER S. 1983. Incremental semantic interpretation in a modular parsing system. In *Automatic natural language parsing*, ed. by Karen Sparck Jones & Yorick Wilks, 148–155. Chichester: Ellis-Horwood.

MEL'ČUK, IGOR A. 1979. *Studies in dependency syntax*. Ann Arbor: Karoma Publishers.

MILLER, GEORGE A. & NOAM CHOMSKY. 1963. Finitary models of language users. In *Handbook of mathematical psychology*, ed. by R. Duncan Luce, Robert R. Bush, & Eugene Galanter, volume II, 419–491. New York: John Wiley.

MILLER, GEORGE A. & PHILLIP N. JOHNSON-LAIRD. 1976. *Language and perception*. Cambridge, MA: Harvard Univ. Press.

MILNE, ROBERT WILLIAM. 1982. Predicting garden path sentences. *Cognitive Science*, 6(4):349–374.

MITCHELL, DON C. & V. M. HOLMES. 1985. The role of specific information about the verb in parsing sentences with local structural ambiguity. *Journal of Memory and Language*, 24:542–559.

MITCHELL, TOM M. 1981. Generalization as search. In *Readings in artificial intelligence*, ed. by Bonnie Lynn Webber & Nils J. Nilsson, 517–542. Los Altos: Morgan Kaufmann.

MONTAGUE, RICHARD. 1973. The proper treatment of quantifiers in ordinary English. In *Formal philosophy*, ed. by Richmond H. Thomason. Yale University Press.

MOORE, ROBERT C. 1989. Unification-based semantic interpretation. In *Proceedings of the 27th Annual Conference of the Association for Computational Linguistics*, 33–41, Vancouver, Canada.

MURPHY, GREGORY L. 1990. Noun phrase interpretation and conceptual combination. *Journal of Memory and Language*, 29:259–288.

NEILL, W. TRAMMELL. 1989. Lexical ambiguity and context: An activation-suppression model. In *Resolving semantic ambiguity*, ed. by David S. Gorfein, 63–83. New York: Springer-Verlag.

NEILL, W. TRAMMELL, D. VANESSA HILLIARD, & ELIZABETH-ANNE COOPER. 1988. The detection of lexical ambiguity: Evidence for context-sensitive parallel access. *Journal of Memory and Language*, 27:279–287.

NG, HWEE TOU & RAYMOND J. MOONEY. 1990. On the role of coherence in abductive explanation. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 337–342. Morgan Kaufmann.

NICOL, JANET & DAVID SWINNEY. 1989. The role of structure in coreference assignment during sentence comprehension. *Journal of Psycholinguistic Research*, 18:5–19.

NORVIG, PETER, 1987. *A unified theory of inference for text understanding*. Berkeley, CA: University of California dissertation. Available as University of California at Berkeley Computer Science Division Technical Report #87/339.

NORVIG, PETER. 1988. Interpretation under ambiguity. In *Proceedings of the 14th Annual Meeting of the Berkeley Linguistics Society*, 188–201, Berkeley, CA.

NORVIG, PETER. 1991. Techniques for automatic memoization with applications to context-free parsing. *Computational Linguistics*, 17(1):91–98.

NORVIG, PETER & GEORGE LAKOFF. 1987. Taking: A study in lexical network theory. In *Proceedings of the 13th Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA.

NORVIG, PETER & ROBERT WILENSKY. 1990. A critical evaluation of commensurable abduction models for semantic interpretation. In *Proceedings of the 13th International Conference on Computational Linguistics*, 225–230, Helsinki.

ODEN, GREGG C. & JAMES L. SPIRA. 1983. Influence of context on the activation and selection of ambiguous word senses. *Quarterly Journal of Experimental Psychology*, 35A:51–64.

OSTERHOUT, LEE & DAVID A. SWINNEY. 1989. On the role of the simplicity heuristic in language processing: Evidence from structural and inferential processing. *Journal of Psycholinguistic Research*, 18(6):553–562.

PEARLMUTTER, NEAL J. & MARYELLEN C. MACDONALD, 1991. Plausibility effects in syntactic ambiguity resolution. Submitted.

PERCIVAL, W. KEITH. 1976. On the historical source of immediate constituent analysis. In *Syntax and semantics volume 7, Notes from the linguistic underground*, ed. by James D. McCawley, 229–242. New York: Academic.

PEREIRA, FERNANDO C. N. 1985. Characterization of attachment preferences. In *Natural language parsing*, ed. by David R. Dowty, Lauri Kartunnen, & Arnold Zwicky, 307–319. Cambridge: Cambridge University Press.

PEREIRA, FERNANDO C. N. & MARTHA E. POLLACK. 1991. Incremental interpretation. *Artificial Intelligence*, 50:37–82.

PEREIRA, FERNANDO C. N. & DAVID H. D. WARREN. 1980. Definite clause grammars for language analysis— a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13(3):231–278.

PEREIRA, FERNANDO C.N. & STUART M. SHIEBER. 1987. *Prolog and natural-language analysis*, volume 10 of *CSLI Lecture Notes*. Chicago: Chicago University Press.

PHILLIPS, BRIAN & JAMES A. HENDLER. 1982. A message-passing control structure for text understanding. In *Proceedings of the Ninth International Conference on Computational Linguistics*, 307–312. North-Holland.

PICKERING, MARTIN & GUY BARRY. 1991. Sentence processing without empty categories. *Language and Cognitive Processes*, 6(3):229–259.

PINKER, STEVEN. 1989. *Learnability and cognition: The acquisition of argument structure*. Cambridge, MA: MIT Press.

POLLARD, CARL & IVAN A. SAG. 1987. *Information-based syntax and semantics: Volume 1: Fundamentals*. Chicago: University of Chicago Press.

POTTER, MARY C. & BARBARA A. FAULCONER. 1979. Understanding noun phrases. *Journal of Verbal Learning and Verbal Behavior*, 18:509–521.

PRATHER, P. A. & DAVID A. SWINNEY. 1988. Lexical processing and ambiguity resolution: An autonomous process in an interactive box. In *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, ed. by Steven L. Small, Garrison W. Cottrell, & Michael K. Tanenhaus, 289–310. San Mateo, CA: Morgan Kaufmann.

PRITCHETT, BRADLEY. 1988. Garden path phenomena and the grammatical basis of language processing. *Language*, 64(3):539–576.

PROUDIAN, DEREK & CARL POLLARD. 1985. Parsing head-driven phrase structure grammar. In *Proceedings of the 23rd Annual Conference of the Association for Computational Linguistics*, 167–171, Chicago.

QUILLIAN, M. ROSS. 1968. Semantic memory. In *Semantic information processing*, ed. by Marvin Minsky, 227–270. Cambridge, MA: MIT Press.

QUIRK, RANDOLPH, SIDNEY GREENBAUM, GEOFFREY LEECH, & JAN SVARTVIK. 1972. *A grammar of contemporary English*. New York: Seminar Press.

REDER, LYNNE M. 1983. What kind of pitcher can a catcher fill? Effects of priming in sentence comprehension. *Journal of Verbal Learning and Verbal Behaviour*, 22:189–202.

REICH, PETER A. 1969. The finiteness of natural language. *Language*, 45(4):831–843.

RIESBECK, CHRISTOPHER K. 1986. From conceptual analyzer to direct memory access parsing: An overview. In *Advances in cognitive science 1*, 236–258. Chichester: Ellis Horwood.

RIESBECK, CHRISTOPHER K. & ROGER C. SCHANK. 1978. Comprehension by computer: Expectation-based analysis of sentences in context. In *Studies in the perception of language*, ed. by W. J. M. Levelt & G. B. Flores d'Arcais, 247–293. London: Wiley.

SADOCK, JERROLD M. 1984. The poly-redundant lexicon. In *Papers from the Parasession on Lexical Semantics*, ed. by David Testen, Veena Mishra, & Joseph Drogo, 250–269, Chicago. Chicago Linguistic Society.

SADOCK, JERROLD M. 1987. Discontinuity in autolexical and autosemantic syntax. In *Syntax and semantics 20: Discontinuous constituency*, ed. by Geoffrey J. Huck & Almerindo E. Ojeda, 283–301. Academic Press.

SALASOO, AITA & DAVID B. PISONI. 1985. Interaction of knowledge sources in spoken word identification. *Journal of Memory and Language*, 24:210–231.

SAMPSON, GEOFFREY. 1980. *Making sense*. Oxford: Oxford University Press.

SAUSSURE, FERDINAND DE. 1915/1966. *Course in general linguistics*. McGraw-Hill. Translated by Wade Baskin.

SCHANK, ROGER C. & ROBERT P. ABELSON. 1977. *Scripts, plans, goals and understanding*. Hillsdale, NJ: Erlbaum.

SCHANK, ROGER C., MICHAEL LEBOWITZ, & LAWRENCE BIRNBAUM. 1980. An integrated understander. *American Journal of Computational Linguistics*, 6(1):13–30.

SCHUBERT, LENHART K. 1984. On parsing preferences. In *Proceedings of the Tenth International Conference on Computational Linguistics*, 247–250.

SCHUBERT, LENHART K. 1986. Are there preference trade-offs in attachment decisions? In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 601–605. Morgan Kaufmann.

SCHUBERT, LENHART K. & FRANCIS J. PELLETIER. 1982. From English to logic: context-free computation of 'conventional' logical translation. *American Journal of Computational Linguistics*, 8(1):27–44.

SCHVANEVELDT, R. W., D. E. MEYER, & C. A. BECKER. 1976. Lexical ambiguity, semantic context, and visual word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 2:243–256.

SEIDENBERG, MARK S., MICHAEL K. TANENHAUS, JAMES M. LEIMAN, & MARIE BIENKOWSKI. 1982. Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. *Cognitive Psychology*, 14:489–537.

SHAPIRO, L. P., E. ZURIF, & J. GRIMSHAW. 1987. Sentence processing and the mental representation of verbs. *Cognition*, 27:219–246.

SHEIL, BEAU A. 1976. Observations on context free parsing. *SMIL: Statistical Methods in Linguistics*, 1:71–109.

SHIEBER, STUART M. 1983. Sentence disambiguation by a shift-reduce parsing technique. In *Proceedings of the 21st Annual Conference of the Association for Computational Linguistics*, 113–118, Cambridge, MA.

SHIEBER, STUART M. 1985. Using restriction to extend parsing algorithms for complex-feature-based formalisms. In *Proceedings of the 23rd Annual Conference of the Association for Computational Linguistics*, 145–152, Chicago.

SHIEBER, STUART M. 1986. *An introduction to unification-based approaches to grammar*. Chicago: University of Chicago Press.

SHIEBER, STUART M., SUSAN U. STUCKY, HANS USZKOREIT, & JANE J. ROBINSON. 1983. Formal constraints on metarules. In *Proceedings of the 21st Annual Conference of the Association for Computational Linguistics*, 22–27, Cambridge, MA.

SHILLCOCK, RICHARD. 1990. Lexical hypotheses in continuous speech. In *Cognitive models of speech processing*, ed. by Gerry T. M. Altmann, 24–49. Cambridge, MA: MIT Press.

SIMPSON, GREG B. 1984. Lexical ambiguity and its role in models of word recognition. *Psychological Bulletin*, 96(2):316–340.

SIMPSON, GREG B. & CURT BURGESS. 1985. Activation and selection processes in the recognition of ambiguous words. *Journal of Experimental Psychology: Human Perception and Performance*, 11(1):28–39.

SIMPSON, GREG B. & CURT BURGESS. 1988. Implications of lexical ambiguity resolution for word recognition and comprehension. In *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, ed. by Steven L. Small, Garrison W. Cottrell, & Michael K. Tanenhaus, 271–288. San Mateo, CA: Morgan Kaufmann.

SIMPSON, GREG B. & GEORGE KELLAS. 1989. Dynamic contextual processes and lexical access. In *Resolving semantic ambiguity*, ed. by David S. Gorfein, 40–56. New York: Springer-Verlag.

SIMPSON, JANE & JOAN BRESNAN. 1983. Control and obviation in Warlpiri. *Natural Language & Linguistic Theory*, 1(3):49–64.

SINGER, MURRAY. 1979. Processes of inference during sentence encoding. *Memory & Cognition*, 7(3):192–200.

SLATOR, BRIAN M. & YORICK WILKS. 1991. PREMO: Parsing by conspicuous lexical consumption. In *Current issues in parsing technology*, ed. by Masaru Tomita, 243–260. Boston: Kluwer.

SMALL, STEVEN L. & CHUCK RIEGER. 1982. Parsing and comprehending with Word Experts. In *Strategies for natural language processing*, ed. by Wendy G. Lehnert & Martin H. Ringle, 89–147. New Jersey: Lawrence Erlbaum.

STABLER, JR, EDWARD P. 1991. Avoid the pedestrian's paradox. In *Principle-based parsing: Computation and psycholinguistics*, ed. by Robert C. Berwick, Steven P. Abney, & Carol Tenny, 199–237. Dordrecht: Kluwer.

STANLEY, RICHARD. 1967. Redundancy rules in phonology. *Language*, 43(2):393–436.

STANNERS, ROBERT F., JAMES NEISER, WILLIAM P. HERNON, & ROGER HALL. 1979. Memory representation for morphologically related words. *Journal of Verbal Learning and Verbal Behavior*, 18:399–412.

STEEDMAN, MARK J. 1989. Grammar, interpretation, and processing from the lexicon. In *Lexical representation and process*, ed. by William Marslen-Wilson, 463–504. Cambridge, MA: MIT Press.

STOWE, LAURIE A., 1984. *Models of gap-location in the human language processor*. Rochester: University of Rochester dissertation. Distributed by the Indiana University Linguistics Club.

STOWE, LAURIE A. 1986. Evidence for on-line gap location. *Language and Cognitive Processes*, 1(3):227–245.

STOWE, LAURIE A. 1989. Thematic structures and sentence comprehension. In *Linguistic structure in language processing*, ed. by G. N. Carlson & M. K. Tanenhaus, 319–357. Dordrecht: Kluwer Academic.

STUCKY, SUSAN U. 1987. Configurational variation in English. In *Syntax and semantics 20: Discontinuous constituency*, ed. by Geoffrey J. Huck & Almerindo E. Ojeda, 377–404. Academic Press.

SWINNEY, DAVID A. 1979. Lexical access during sentence comprehension: (re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, 18:645–659.

SWINNEY, DAVID A. & ANNE CUTLER. 1979. The access and processing of idiomatic expressions. *Journal of Verbal Learning and Verbal Behavior*, 18:523–534.

SWINNEY, DAVID A. & LEE OSTERHOUT. 1990. Inference generation during auditory language comprehension. In *Inferences and text comprehension*, ed. by Arthur C. Graesser & Gordon H. Bower. Academic Press.

TABOSSI, PATRIZIA. 1988. Sentential context and lexical access. In *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, ed. by Steven L. Small, Garrison W. Cottrell, & Michael K. Tanenhaus, 331–342. San Mateo, CA: Morgan Kaufmann.

TALMY, LEONARD. 1977. Rubber-sheet cognition in language. In *Papers from the 13th Regional Meeting of the Chicago Linguistics Society*, Chicago.

TALMY, LEONARD. 1978. The relation of grammar to cognition – a synopsis. In *Proceedings of TINLAP-2 (Theoretical Issues in Natural Language Processing)*, ed. by David Waltz, Champaign, IL. Coordinated Science Laboratory, University of Illinois.

TALMY, LEONARD. 1986. The relation of grammar to cognition. Technical Report 45, Berkeley Cognitive Science Report, Berkeley.

TANENHAUS, MICHAEL K., JULIE E. BOLAND, SUSAN M. GARNSEY, & GREG CARLSON. 1989. Lexical structures in parsing long-distance dependencies. *Journal of Psycholinguistic Research*, 18(1):37–50.

TANENHAUS, MICHAEL K. & GREG CARLSON. 1989. Lexical structure and language comprehension. In *Lexical representation and process*, ed. by William Marslen-Wilson, 529–561. Cambridge, MA: MIT Press.

TANENHAUS, MICHAEL K., JAMES M. LEIMAN, & MARK S. SEIDENBERG. 1979. Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior*, 18:427–440.

TANENHAUS, MICHAEL K. & MARGERY M. LUCAS. 1987. Context effects in lexical processing. *Cognition*, 25:213–234.

TANENHAUS, MICHAEL K., LAURIE A. STOWE, & GREG CARLSON. 1985. The interaction of lexical expectation and pragmatics in parsing filler-gap constructions. In *Program of the Seventh Annual Conference of the Cognitive Science Society*, 361–365, Irvine, CA.

TARABAN, ROMAN & JAMES L. MCCLELLAND. 1988. Constituent attachment and thematic role assignment in sentence processing: Influences of content-based expectations. *Journal of Memory and Language*, 27:597–632.

TESNIÈRE, LUCIEN. 1959. *Éléments de syntaxe structurale*. Paris: Librairie C. Klincksieck.

THOMPSON, HENRY S., MIKE DIXON, & JOHN LAMPING. 1991. Compose-reduce parsing. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*, 87–97.

TOMITA, MASARU. 1987. An efficient augmented context-free parsing algorithm. *Computational Linguistics*, 13(1–2):31–46.

TRUESWELL, JOHN C. & MICHAEL K. TANENHAUS. 1991. Tense, temporal context and syntactic ambiguity resolution. *Language and Cognitive Processes*, 6(4):303–338.

TYLER, LORRAINE K. 1984. The structure of the initial cohort: Evidence from gating. *Perception & Psychophysics*, 36(5):417–427.

TYLER, LORRAINE K. 1989. The role of lexical representations in language comprehension. In *Lexical representation and process*, ed. by William Marslen-Wilson, 439–462. Cambridge, MA: MIT Press.

TYLER, LORRAINE KOMISARJEVSKY & WILLIAM MARSLEN-WILSON. 1982. Speech comprehension processes. In *Perspectives on mental representation.*, ed. by Jacques Mehler, Edward C. T. Walker, & Merrill Garrett, 169–184. Hillsdale, NJ: Erlbaum.

ULVESTAD, BJARNE. 1960. On the use of transitional probability estimates in programming for mechanical translation. *Statistical Methods in Linguistics*, 1:24–40.

USZKOREIT, HANS. 1986. Categorial unification grammars. In *Proceedings of the 11th International Conference on Computational Linguistics*, Bonn. University of Bonn.

VAN DER LINDEN, ERIC-JAN & WESSEL KRAAIJ. 1990. Ambiguity resolution and the retrieval of idioms: two approaches. In *Proceedings of the 13th International Conference on Computational Linguistics*, 245–250, Helsinki.

VAN NOORD, GERTJAN. 1991. Head corner parsing for discontinuous constituency. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*, 114–121.

VAN PETTEN, CYMA & MARTA KUTAS. 1988. Tracking the time course of meaning activation. In *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, ed. by Steven L. Small, Garrison W. Cottrell, & Michael K. Tanenhaus. San Mateo: Morgan Kaufmann.

VENNEMANN, THEO. 1974. Words and syllables in natural generative grammar. In *Papers from the Parasession on Natural Phonology*, 346–374, Chicago. Chicago Linguistic Society.

WALTZ, DAVID L. & JORDAN B. POLLACK. 1985. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9(1):51–74.

WANNER, ERIC. 1980. The ATN and the Sausage Machine: Which one is baloney. *Cognition*, 8:209–226.

WARD, NIGEL, 1991. *A flexible, parallel model of natural language generation*. Berkeley, CA: University of California dissertation. Available as University of California at Berkeley Computer Science Division Technical Report #UCB/CSD/91/629.

WELLS, RULON S. 1947. Immediate constituents. *Language*, 23(2):81–117.

WEST, MICHAEL. 1953. *A general service list of English words*. London: Longmans.

WHITTEMORE, GREG, KATHLEEN FERRARA, & HANS BRUNNER. 1990. Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th Annual Conference of the Association for Computational Linguistics*, 23–30, Pittsburgh, PA.

WILENSKY, ROBERT. 1983. *Planning and understanding*. Reading, MA: Addison-Wesley.

WILENSKY, ROBERT. 1986. Some problems and proposals for knowledge representation. In *Experience, memory, and reasoning*, ed. by Janet L. Kolodner & Christopher K. Riebeck, 15–28. New Jersey: Lawrence Erlbaum.

WILENSKY, ROBERT. 1990. Extending the lexicon by exploiting subregularities. In *Proceedings of the 13th International Conference on Computational Linguistics*, 407–412, Helsinki.

WILENSKY, ROBERT & YIGAL ARENS. 1980. PHRAN: A knowledge-based natural language understander. In *Proceedings of the 18th Annual Conference of the Association for Computational Linguistics*, Philadelphia.

WILKS, YORICK. 1975a. An intelligent analyzer and understander of English. *Communications of the ACM*, 18(5):264–274.

WILKS, YORICK. 1975b. Preference semantics. In *The formal semantics of natural language*, ed. by Edward L. Keenan, 329–350. Cambridge: Cambridge Univ. Press.

WILKS, YORICK. 1975c. A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, 6(1):53–74.

WILKS, YORICK, XIUMING HUANG, & DAN FASS. 1985. Syntax, preference and right attachment. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 779–784.

WINOGRAD, TERRY. 1972. *Understanding natural language*. New York: Academic Press.

WOODS, WILLIAM A. 1970. Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10):591–606.

WOODS, WILLIAM A. 1977. Lunar rocks in natural English: Explorations in natural language question answering. In *Linguistic structures processing*, ed. by Antonio Zampolli, 521–569. Amsterdam: North Holland.

WRIGHT, BARTON & MERRILL GARRETT. 1984. Lexical decision in sentences: Effects of syntactic structure. *Memory & Cognition*, 12(1):31–45.

WU, DEKAI. 1990. Probabilistic unification-based integration of syntactic and semantic preferences for nominal compounds. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki.

WU, DEKAI, 1992. *Automatic inference: A probabilistic basis for natural language interpretation*. Berkeley, CA: University of California dissertation.

YNGVE, VICTOR H. 1955. Syntax and the problem of multiple meaning. In *Machine translation of languages*, ed. by William N. Locke & A. Donald Booth. Cambridge, MA: MIT Press.

ZIMMER, KARL. 1964. *Affixal negation in English and other languages: An investigation of restricted productivity*. Supplement to *Word*, Monograph 5. New York: .

ZWICKY, ARNOLD. 1978. Arguing for constituents. In *Papers from the 14th Regional Meeting of the Chicago Linguistics Society*, 503–512, Chicago.

ZWICKY, ARNOLD. 1987. Constructions in monostratal syntax. In *Papers from the 23rd Annual Regional Meeting of the Chicago Linguistics Society*, ed. by Barbara Need, Eric Schiller, & Anna Bosch, 389–401, Chicago.

ZWICKY, ARNOLD. 1989. What's become of derivations? Defaults and invocations. In *Proceedings of the 15th Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA.

ZWITSERLOOD, PIENIE. 1989. The locus of the effects of sentential-semantic context in spoken-word processing. *Cognition*, 32:25–64.