Grammar, Uncertainty and Sentence Processing

by

John T. Hale

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

August, 2003

© John T. Hale 2003

All rights reserved
Abstract

Toward a probabilistic theory of human sentence processing, this dissertation proposes a definition of computational work done in the course of analyzing sentences generated by formal grammars. It applies the idea of entropy from information theory to the set of derivations compatible with an initial substring of a sentence. Given a probabilistic grammar, this permits the set of such compatible derivations to be viewed as a random variable, and the change in uncertainty about the outcomes to be calculated.

This definition of computational work is examined as a cognitive model of human sentence processing difficulty. To apply the model, a variety of existing syntactic proposals for English sentences are cast as probabilistic Generalized Phrase Structure Grammars (Gazdar et al., 1985) and probabilistic Minimalist Grammars (Stabler, 1997). It is shown that the amount of predicted processing effort in relative clauses correlates with the Accessibility Hierarchy of relativized grammatical relations (Keenan and Comrie, 1977) on a Kaynian (1994) view of relative clause structure. Results from three new sentence reading experiments confirm the findings of Keenan and Hawkins (1987) by demonstrating effects of the Accessibility Hierarchy on question-answering accuracy, but find only limited support for the AH in online reading times. These latter results suggest that while genitivity and obliqueness make relative clause processing harder, indirect object extraction is not significantly harder than direct object extraction.

Advisor: Paul Smolensky
Readers: William Badecker, Edward Gibson,
          Scott Smith, Edward P. Stabler, Jr.
Acknowledgements

I wish to sincerely thank my committee: Paul Smolensky, Ed Stabler, Ted Gibson, Bill Badecker and Scott Smith, my teachers: Eugene Charniak, Pauline Jacobson, Ed Keenan and Larry Moss, comrades in the grad student experience especially my incoming cohort: Lisa Davidson, Nate Greenslit and Diana Schenck, but also: Matt Goldrick, Paul Hagstrom, Jussi Valtonen and Julia Hockenmaier, and the cogsci department staff: Bernadette Butler, Barb Fisher, Don Mathis, Sue Potterfield, and Rachel Sala. CS posse: Gideon Mann, Charles Schafer and Chris Skalka, LA crew: Colin Wilson, Eric Baković, Henk Harkema, Greg Kobele, Jason Riggle, Heriberto Avelino and Sahyang Kim, Boston society, including the corrections: Mario Corsetti, Jon Soule and Duane Watson, as well as TedLabbers: Dan Grodner, Doug Rohde and Florian Wolf, Baltimoreans: Jan Danek, Paul Ballard and all choristers, parishioners and clergy at Corpus Christi, life-altering friends: Tahir Butt, Gitana Chunyo, Dave Goldberg, Jenny Hunt, Alexis Lewis, Victor Osinaga, Mike Saddoris, deep-structural family: Tom Lalinsky and Judy McMenamin, and most importantly my parents, Thomas and Barbara Hale for literally everything.

This dissertation is dedicated to my deceased grandparents Albert G. & Constance Y. L. Hale and Emil & Mary Lalinsky.
Contents

Abstract ii
Acknowledgements iii
Contents iv
List of Figures vii

1 Introduction 1
  1.1 The role of uncertainty in
       a competence-performance hypothesis 2
  1.2 The structure of the argument 2
  1.3 The structure of this dissertation 3

2 Entropy reduction, context-free grammar and human sentence processing 4
  2.1 Introduction 4
  2.2 The Terrain 5
  2.3 Sentence Processing as Entropy Reduction 7
     2.3.1 Probabilistic Context-Free Grammars 8
     2.3.2 Entropy 9
     2.3.3 Conditional entropy 12
     2.3.4 Left recursion 13
  2.4 Examples 13
     2.4.1 Main-verb/reduced relative 13
     2.4.2 The NP/S and NP/Z ambiguities 16
     2.4.3 Subject and Object relatives 20
     2.4.4 Center-embedding 23
  2.5 Conclusion 23
  2.6 Appendix: Consistency of PCFGs 24

3 A probabilistic, mildly-context-sensitive hypothesis
   about human sentence processing 26
  3.1 Orientation: which field? 26
     3.1.1 The Competence hypothesis as a methodological principle 27
     3.1.2 Probabilistic grammar 28
  3.2 The Entropy Reduction hypothesis 29
     3.2.1 Entropy of nonterminals in a PCFG 30
     3.2.2 Incomplete sentences 31
     3.2.3 A more specific hypothesis 32
  3.3 The Accessibility Hierarchy 33
6 Kinds of probabilistic sentence processing theories
   6.1 Empirical coverage ........................................ 131
   6.2 Quantity proposed as a psycholinguistic model ........... 131
       6.2.1 Ranking theories ................................. 131
       6.2.2 Other scalar functions of a probabilistic grammar and a prefix string 132
   6.3 Grammar type .......................................... 136

7 Conclusions and Future Work................................. 139
   7.1 Conclusions ........................................... 139
   7.2 Future work .......................................... 140

A Implementing a parser for Minimalist Grammars in a functional programming language .... 141
   A.1 Grammar ............................................. 142
   A.2 Items ................................................ 142
       A.2.1 Variables ....................................... 143
   A.3 The chart ........................................... 146
       A.3.1 Lookup ......................................... 146
       A.3.2 Derivations .................................... 147
   A.4 The agenda .......................................... 149
   A.5 Unification ......................................... 150
   A.6 Speed ................................................. 152
   A.7 Conclusion .......................................... 153

Vita .............................................................. 170
List of Figures

2.1 Example probabilistic context-free phrase structure grammar .......................... 9
2.2 Two derivations of “The cop saw the spy with binoculars” .......................... 9
2.3 Entropy of a fair die  ........................................................................... 10
2.4 Computing the entropy of the partial derivations common to $\mathcal{D}_A$ ........... 12
2.5 Initial Bever Grammar ........................................................................... 14
2.6 Non-left-recursive Bever Grammar ....................................................... 14
2.7 Entropies of nonterminals in Non-left-recursive Bever Grammar ............... 15
2.8 Parser state having heard “the” .......................................................... 15
2.9 Parser state having heard “the horse” ................................................... 15
2.10 Parser state having heard “the horse raced” ......................................... 16
2.11 Bits of entropy reduction for a garden path sentence ............................... 17
2.12 Grammar for NP/S and NP/Z ambiguities ............................................. 19
2.13 PCFG for center-embedding and Grodner et al. (2000) subject-object stimuli 21
2.14 Subject relative (observed data from Grodner et al. (2000)) .................... 22
2.15 Object relative (observed data from Grodner et al. (2000)) .................... 22
2.16 Total information transacted at increasing levels of center-embedding ...... 23
2.17 Inconsistent PCFG .............................................................................. 25

3.1 A very simple PCFG ............................................................................. 29
3.2 Recursive PCFG ................................................................................... 30
3.3 Sentences generated by recursive PCFG ................................................. 30
3.4 “Situated” grammar rule ...................................................................... 32
3.5 The Accessibility Hierarchy of relativizable grammatical relations ............. 33
3.6 results from Keenan & S. Hawkins (1987) ............................................. 35
3.7 A Minimalist Grammar .......................................................................... 41
3.8 Derivation tree and structural description of “the boy who I met” ............. 41
3.9 Kaynian promotion analysis ................................................................... 44
3.10 more standard adjunction analysis ....................................................... 44
3.11 Results of ‘Variation in UG’ corpus study, $N = 2238$ ............................... 46
3.12 Interpolated weights used to train PCFG ................................................. 47
3.13 Derivation-PCFG for Keenan and S. Hawkins test set ............................. 48
3.14 Entropy reduction vs. position on AH .................................................... 49
3.15 Entropy reduction vs. subjects’ error scores ............................................ 50
3.16 Log probabilities vs. position on AH ...................................................... 50
3.17 Entropy reductions on adjunction-analysis grammar vs. position on AH ... 51
3.18 Entropy reduction on a uniformly-weighted grammar vs. position on AH ... 51
3.19 Derivation tree size vs. position on AH ............................................... 52
3.20 Word-by-word entropy reductions for subject extraction ....................... 53
3.21 Word-by-word entropy reductions for DO-extraction ............................. 54
3.22 Word-by-word entropy reductions for IO-extraction .......................... 55
3.23 Word-by-word entropy reductions for oblique-extraction ...................... 56
3.24 Word-by-word entropy reductions for genitive subject-extraction .......... 57
3.25 Word-by-word entropy reductions for genitive object-extraction .......... 58
3.26 Assumed GPSG-style analysis of initial part of DO stimulus .................. 65
3.27 Assumed GPSG-style analysis of initial part of IO stimulus .................. 65
3.28 grammar of VP/NP ....................................................................... 67
3.29 grammar of VP/PP ....................................................................... 68

4.1 Stimuli from Hawkins and Keenan 1974/1987 .......................................... 94
4.2 IO merged first, then DO, then Subject ............................................. 94
4.3 Simplified promotion analysis of an object-extracted relative clause .......... 95
4.4 extraction from genitive subject: who + ’s → whose ............................... 95
4.5 unshifted dative ............................................................................. 96
4.6 shifted dative ............................................................................... 96
4.7 Some auxiliaries ........................................................................... 97
4.8 VP modifier ................................................................................ 97
4.9 configurational definitions of grammatical relations ............................ 98

5.1 Experiment 1 average word-by-word reading times ................................. 103
5.2 Experiment 2 average word-by-word reading times ................................. 105
5.3 Experiment 3 average regional reading times ....................................... 107

6.1 Object relative clause ..................................................................... 133
6.2 A grammar rule for proper names ...................................................... 137

A.1 How unification is used for each case of parsing-as-deduction ............... 151
A.2 Subsumption testing in the queue .................................................... 152
A.3 Unification success rates ................................................................. 153
A.4 Parsing sentences in Accessibility Hierarchy test suite .......................... 154
Chapter 1

Introduction

The I-language is a (narrowly described) property of the brain, a relatively stable element of transitory states of the language faculty. Each linguistic expression (SD) generated by the I-language includes instructions for performance systems in which the I-language is embedded. It is only by virtue of its integration into such performance systems that this brain state qualifies as language. [emphasis added]

(Chomsky, 1992, 213)

Since its inception in the 1950s, cognitive science has proceeded from the assumption that certain kinds of thinking are like the kinds of information-processing that computers do. This view would not be revolutionary if it did not entail an agenda of showing that non-obviously computational sorts of thinking can, in fact be given a computational explanation. This is also the agenda of this dissertation: to explain an aspect of a particular kind of cognitive behavior in terms of information processing. The cognitive behavior is language understanding – whatever mental activity it is that occurs in between hearing (or reading) a sentence and being able to accurately answer questions about what that sentence means. The aspect under consideration is the relative difficulty of this task. Why is it that some sentences, or even specific words within sentences, are more difficult to understand than others?

In order to answer this question, some position must be taken on exactly what work people are doing as they understand sentences. In this dissertation, I assume that one kind of work being done is the determination of the linguistic structure of the sentence being understood. This follows Chomsky (1965) in assuming that the same knowledge of grammar uncovered in linguistics ought to be a key part of an account of language use. While this view is uncontroversially accepted in some circles, there is no consensus about the precise relationship between such knowledge of language and its use in sentence processing.

This dissertation defines an explicit relationship between grammars and their use by abstract computers – possibly including people. It shows how to use this relationship to
derive predictions about sentence understanding difficulty, and compares these predictions with existing and novel empirical findings. In this regard, it contributes to cognitive science by characterizing a computation that can be offered as a psycholinguistic hypothesis.

1.1 The role of uncertainty in a competence-performance hypothesis

The proposal is that sentence processing work is disambiguation work. It is motivated by a view of the human sentence understander as an optimizer that uses a probabilistic grammar to constantly hedge bets about possible structures for a given sentence. In the course of word-by-word sentence understanding, grammatical possibilities are successively ruled out, and the processor’s portfolio is re-adjusted to accommodate the new set of contenders. When a word takes the processor from a state of relative indifference to one of more localized suspicion, comparatively more disambiguation has occurred. This is plausible on the view of sentence understanding as a process of increasing certainty about the particular analysis of a grammatical sentence intended by the speaker. The formalization of “indifference” and “localized suspicion” is in terms of the notion of entropy from information theory in the sense of Shannon (1948). This theory presupposes a probability space of events and hence requires the adoption of some form of probabilistic grammar.

1.2 The structure of the argument

The thesis proposed in this dissertation is that there exists an information-theoretic notion of disambiguation work on a probabilistic grammar. Immediately this makes possible the cognitive hypothesis that the scale of human sentence processing difficulty is well-modeled by this formal definition of disambiguation work. Call this cognitive hypothesis \( H_c \), a hypothesis about how comprehenders compute using grammars. In order to apply it in any specificity, it needs to be combined with a hypothesis \( H_g \) about the grammars being computed over. In this dissertation, \( H_g \) is just a grammar fragment for some language. The domain of these two ideas is the realm of sentence understanding behaviors: \( P_{bb} \). Following Wirth (1975) the structure of the argument can be depicted in 1.1.

\[
H_g \land H_c \rightarrow P_{bb}
\]  

(1.1)

\(^1\)The conception of probabilistic grammar used in this thesis is the one that is commonly adopted in computational linguistics (Charniak, 1993, chapter 5) (Jurafsky and Martin, 2000, chapter 12) (Manning and Schütze, 2000, chapter 11). It has also been presented from a psycholinguistic point of view (Levett, 1974, chapter 3). A general theory of alternative formulations of probabilistic grammar is developed in Sannéedsson (2000).
Assuming, with Chomsky, that linguistic grammars are in fact implicated in sentence understanding, Wirth (1975) points out that it is the conjunction of particular hypotheses about grammar and computation that jointly yield predictions about observable sentence understanding behavior. This means that $P_{4b}$ being actually borne out experimentally confirms only the conjunction $H_{g} \land H_{c}$, and not one or the other conjuncts in isolation. Since the thesis proposed in this dissertation is about computations over a grammar (i.e. that a notion of ‘work’ can be defined on them) grammar fragments that are well-established on purely linguistic grounds will be used to instantiate $H_{g}$. Making $H_{g}$ as strong as possible in this way will make any verifiable predictions as informative as possible about the truth of $H_{c}$.

1.3 The structure of this dissertation

The dissertation argues for $H_{c}$ – the claim that an information-theoretic definition of disambiguation work is a valuable cognitive model – primarily by example. In chapter 2 the first example is given, where various $H_{g}$ in the class of context-free grammars are explored. These miniature grammars are written in the style of Generalized Phrase Structure Grammar (Gazdar et al., 1985). There $H_{c}$ is defined more precisely, and a procedural method for calculating the joint predictions of the two hypothesis is given. These predictions are compared with existing experimental evidence, including on-line reading times. Chapter 3 generalizes and refines these proposals with a different example. It presents another way of calculating behavioral predictions that extends to grammar formalisms more expressive than context-free grammars. A particular grammar written in the Minimalist Grammars (MG) formalism of Stabler (1997) is used to make predictions about the relative understandability of a hierarchy of sentence types discovered by Keenan and Comrie (1977). These predictions correlate with repetition accuracy results obtained by Keenan and Hawkins (1987), but only on a grammar incorporating Kayne’s (1994) proposal about relative clause structure. This and other linguistic proposals inherent in the grammar are discussed in chapter 4, which shows how the MG formalism can encode heterogenous ideas from many schools of linguistics. Motivated by the candidate theory described in chapter 3, chapter 5 presents the results of three sentence understanding experiments. It finds that the predictions of the Keenan-Comrie (1977) hierarchy are only partially upheld in reading times recorded during online sentence understanding. This demonstrates how a new theoretical proposal can lead to a new investigation of empirical phenomena that revises our previous understanding. Finally, chapter 6 situates the technical proposals of chapters 2 and 3 in a space of possible probabilistic sentence processing theories. Chapter 7 draws conclusions and identifies future work.
Chapter 2

Entropy reduction, context-free grammar and human sentence processing

2.1 Introduction

Ambiguity resolution is perhaps the central problem (Tabor and Tanenhaus, 2001) in sentence processing. How is it that human sentence understanders are able to recognize combinatory relationships, from an infinite range of possibilities, to arrive at a meaningful interpretation of a spoken or written sentence?

This question is typically addressed experimentally, using some measure of cognitive load to reason backwards to the choices a sentence understander has made in online processing. The present chapter inverts this usual arrangement by showing how a fairly general conception of ambiguity resolution is sufficient to characterize a range of cognitive load patterns. Ambiguity resolution in this sense is the elimination of impossible structural analyses for a string – a natural measure of the cognitive ‘work’ a comprehender does on the way to determining the speaker’s intended analysis.

This demonstration implies that some aspects of cognitive load as measured in reading time experiments follow simply from the statement of the sentence understanding problem as an ambiguity resolution problem. It also suggests a way that explicit linguistic knowledge – both symbolic combinatory knowledge as well as numerical knowledge expressed as probabilities – can be used by comprehenders. In doing so, it adopts the competence hypothesis (Chomsky, 1965) that our knowledge of language is directly used in comprehension.

An earlier revision of this chapter appears in the volume 32, number 2 of the Journal of Psycholinguistic Research (March 2003).
The main claim is that cognitive load is related, perhaps linearly, to the reduction in the perceiver’s uncertainty about what the producer meant. Historical antecedents and a few closely related proposals are examined in a brief section immediately following this introduction. The central section formalizes the main claim by appealing to information theory, in a way that avoids the criticisms of Chomsky (1956), by using a phrase-structured language model. The penultimate section shows how the claim derives processing predictions about constructions that have been well studied in the sentence processing literature. The final section concludes with some speculations about connectionist networks whose operation accord with the main claim.

2.2 The Terrain

Psycholinguistic research on human syntactic processing has always sought to integrate a wide variety of findings into explicit, general theories. This kind of synthesis, toward which the current chapter strives, is often very difficult because of the range of natural language phenomena a processing theory must confront.

For instance, in empirical work over the past two decades, evidence has been accruing that the human sentence processor is sensitive to gradient factors like frequency (MacDonald, Pearlmutter, and Seidenberg, 1994; Trueswell, 1996; Mitchell et al., 1995) thematic fit (Trueswell, Tanenhaus, and Garnsey, 1994; Garnsey et al., 1997) and pragmatic felicity (Tanenhaus et al., 1995). These findings are reviewed by Tanenhaus and Trueswell (1995) and Gibson and Pearlmutter (1998). At the same time, there has been a corresponding revival of interest on the theoretical side in probabilistic grammars as explanations for these effects, as well as core phenomena not previously viewed as gradient.

Although strongly motivated by recent findings, the idea of using probabilistic grammars in psycholinguistic research is an old one, going back to the work of Patrick Suppes. Suppes (1970) proposes the specific formulation of probabilistic grammar used in this chapter as an account of the child’s developing knowledge of language. He points out that since probabilistic grammars define the frequencies of words and phrases in a language, they are like any other statistical model whose fit to a given sample can be evaluated in a standardized way. This view, adopted in the present work, takes probabilities to be linguistic properties that are predicted by a grammar, on a par with acceptability distinctions.

Explicit mechanisms of probabilistic processing have been pursued most intently by connectionists. In the PDP tradition (McClelland and Kawamoto, 1985; McClelland and John, 1989; St. John and McClelland, 1990) it has long been the norm to view constraints on sentence processing, if not grammar itself (Legendre, Miyata, and Smolensky, 1990b; Legendre, Miyata, and Smolensky, 1990a) as being partial and numerically-valued.
This view takes on a new form in more recent work (Elman, 1990; Tabor, Juliano, and Tanenhaus, 1997; Rohde, 2002) where back-propagation is applied to learning the string sets and frequencies of given grammars. A drawback to this approach is that the internal states of the induced machines are often uninterpretable (Steedman, 1999). This criticism takes on a diminished relevance as the application of dynamical systems theory to these states becomes more and more refined (Rodriguez, 1999; Tabor, 2000).

At issue more centrally are the kinds of general principles that (perhaps imperfectly) characterize the operation of sentence processing models estimated from linguistic data.

For example, on the Visitation Set Gravitation model of Tabor, Juliano, and Tanenhaus (1997) reading times can be derived from a post-hoc analysis of a trained Simple Recurrent Network (SRN). This analysis yields a landscape of attractors from the records of hidden unit activations. By observing how long it takes a particular hidden unit state (representing a word along with its left-context) to ‘gravitate’ into an attractor (possibly representing a kind of semantic integration), Tabor and colleagues obtain a measure of the work a comprehender does integrating a word into a developing analysis.

In these experiments, it can take longer to settle on an attractor when many competing attractors – parser states – need to be considered. This is a joint consequence of the gravitational parameters, and the localization, by back-propagation, of similar parser states nearby one another. It may be that high predicted reading times are the result of the parser traversing ‘confusing’ regions of hidden unit space where the influence of many attractors is simultaneously felt. This supposition motivates the development, in the next section, of a notion of uncertainty or confusion that might characterize, at a high level, what the Visitation Set Gravitation model is doing.

An independent line of work (Jurański, 1996; Narayanan and Jurański, 1998; Narayanan and Jurański, 2001) with Bayesian nets (Pearl, 1988; Jensen, 1996) takes a more strongly grammatical approach in the tradition of localist connectionism (Feldman and Ballard, 1982). A Bayesian net is a kind of graphical model that can be hand-crafted to represent a distribution over linguistic variables, such as constructions or lexical entries. Each node of the net is associated with a conditional probability table expressing a distribution on values of the node’s variable. For instance, a constituent like NP (noun phrase) might be associated with a distribution over \{true, false\} encoding the net’s belief that the constituent is really present or not. The edges between nodes have a conditional probability interpretation, so they can be estimated directly from large corpora. Evidence is inserted into a Bayesian net by setting the state of ‘observable’ nodes and then propagating probabilities across the edges of the graph according to Bayes’ Law.

Jurański and Narayanan examine how the probabilities change as the net’s beliefs
are updated in light of new evidence. The approach is strongly grammatical because the node probabilities can be immediately interpreted in terms of a linguistic analysis. For instance, Jurafsky (1996) discusses the simultaneous consideration of two constructions, the Main-Clause-Non-Subject-Wh-Question and the Subject-Matrix-Wh-Question upon being presented with the initial string “who can...”. If one construction becomes overwhelmingly plausible, due to any combination of evidence, Bayesian belief propagation allows it to ‘explain away’ evidence for the alternatives. The probabilities of individual constructions represented in the Bayes net can go up and down as processing proceeds. Using the idea that the parser pursues only the highest probability analyses (Kurtzman, 1985; Gibson, 1991) the Bayesian approach predicts garden path effects when an analysis falls below some threshold, and is forgotten, but is later required when no other alternative is viable.

Jurafsky and Narayanan’s predictions about human processing derive from probabilities of representations defined by a grammar, so their work respects the competence hypothesis mentioned in the introduction. However, these calculations are carried out over Bayesian nets for complete linguistic analyses. It is not clear if a dynamically-constructed Bayes net (Charniak and Goldman, 1993) would derive the same predictions. At the same time, such ‘partial’ networks would seem to be called for to cover the full set of constructions defined by a recursive grammar. This kind of summation over an infinite number of linguistic representations is implicitly performed by using a closed-form solution of an infinite series (equation 2.6) in the next section to calculate psycholinguistic predictions about cognitive load.

Finally, Den and Inoue (Den and Inoue, 1997; Inoue and Den, 1999) follow Jurafsky in endorsing a beam-search interpretation of garden pathing. In this pioneering application of information theory to sentence processing, the analyses considered by a parallel parser are ranked according to the uncertainty of the distribution of verbs licensed to occur in a particular syntactic configuration. Den and Inoue address the puzzle of comprehenders’ expectations for particular verbs following sequences of candidate arguments in verb-final languages such as Japanese. Their Verb Predictability Model would seem to be a special case of the more general architecture to be proposed in the following section, which is sensitive to the predictability of all structures, not just verbs. However, this proposal differs with Den and Inoue in avoiding any appeal to beam search or reanalysis.

2.3 Sentence Processing as Entropy Reduction

This section shows how to derive processing predictions about reading time from probabilistic grammars encoding linguistic generalizations that are both categorial and nu-
merical. Three simplifying assumptions place general constraints on sentence processing as an ambiguity resolution problem.

1. During comprehension, sentence understanders determine a syntactic structure for the perceived signal.

2. Producer and comprehender share the same grammar.

3. Comprehension is eager; no processing is deferred beyond the first point at which it could happen.

These assumptions suppose that there are combinatory relationships among words presented during incremental sentence processing. A probabilistic grammar is known to both speaker and hearer; the derivations of this grammar completely determine the combinatory relationships to be recognized. Because of the ambiguity of sentence beginnings, however, there is uncertainty about which derivation the speaker intends. This uncertainty is especially great when only an initial segment of a sentence has so far been presented. But if the processor performs disambiguating work, this uncertainty can be expected to go down as more words are presented.

If uncertainty about a derivation measures the total amount of ambiguity-resolution work a processor will have to do, then reduction in this uncertainty should measure the maximal amount of work that can be done between one word and the next. This is the amount of work done by an eager sentence processor.

The reduction in uncertainty from one word to the next is the information conveyed by that word. To the extent that sentence comprehension is eager, this information conveyed should closely match other word-by-word measures of information processing.

2.3.1 Probabilistic Context-Free Grammars

I will adopt the formalism of probabilistic (or stochastic) context-free grammars (PCFGs) to make the discussion of derivations on a grammar explicit. More complete presentations can be found in all modern computational linguistics texts (Charniak, 1993; Jurafsky and Martin, 2000; Manning and Schütze, 2000). Intuitively, a PCFG is just the kind of phrase structure grammar familiar from linguistics, augmented with probabilities on the rules. The rules of an example grammar are given in Figure 2.1.

The rules in Figure 2.1 have the property that the probabilities of all rules expanding the same nonterminal symbol sum to 1. Grammars with this property are 'normalized' which, by itself, is not enough to guarantee a proper probability model (see Appendix). Figure 2.1 also shows how the idea of derivational ambiguity carries over from formal language theory.
1.0  S  →  NP VP  
0.4  NP  →  the spy  
0.4  NP  →  the cop  
0.2  NP  →  NP PP  
1.0  PP  →  with binoculars  
0.6  VP  →  saw NP  
0.4  VP  →  saw NP PP  

Figure 2.1: Example probabilistic context-free phrase structure grammar

S  
NP VP  
the cop VP  
the cop saw NP  
the cop saw NP PP  
the cop saw the spy PP  
the cop saw the spy with binoculars

Figure 2.2: Two derivations of “The cop saw the spy with binoculars”

These rules generate the string “the cop saw the spy with binoculars” by two different derivations, given in Figure 2.2.

Note first that all derivations begin with the start symbol S. The derivation on the right side of Figure 2.2 has probability $(0.4)^3 = 0.064$. The derivation on the left side of Figure 2.2 has probability $(0.4)^2 \times 0.6 \times 0.2 = 0.0192$. The probability of a generated string is just the sum of the probabilities of all derivations that generate it: $0.0192 + 0.064 = 0.0832$. The idea of sentence processing as entropy reduction is that all of the work that the processor does is like the kind of ambiguity resolution needed to decide between these two derivations, and that the magnitude of this work is measured by reading time.

### 2.3.2 Entropy

Ambiguity-resolution work can now be formalized as the information conveyed by a word in a sentence generated by a PCFG. Essential to the definition of information conveyed is the notion of entropy (Shannon, 1948). The entropy of a random variable is the uncertainty, or missing information, associated with that random variable. More explicitly, for a discrete random variable $X$ with outcomes $x_1, x_2, \ldots$ having probabilities $p_{x_1}, p_{x_2}, \ldots$ the entropy $H(X)$ is

$$H(X) = - \sum_{x \in X} p_x \log_2 p_x \quad (2.1)$$

The form of equation 2.1 shows that entropy is identical with the expected surprisal $\log_2 p_x$ of an unknown outcome. Different distributions on the $p_{x_i}$ lead to different entropies,
$DIE =\begin{array}{c|c}
\text{value} & \text{probability} \\
\hline
1 & \frac{1}{6} \\
2 & \frac{1}{6} \\
3 & \frac{1}{6} \\
4 & \frac{1}{6} \\
5 & \frac{1}{6} \\
6 & \frac{1}{6} \\
\end{array}$

$p_{DIE = 3} = \frac{1}{6}$

$H(DIE) = - \sum_{x \in DIE} p_x \log_2 p_x$

$\approx 2.58\text{bits}$

Figure 2.3: Entropy of a fair die

measured in bits when the base of the logarithm is 2. When all outcomes are equally likely, entropy is maximal. For example, the entropy of a fair die is about two and a half bits (Figure 2.3).

Under the assumption that sentence understanders determine a syntactic structure for the sentences being understood, the random variable of interest must be one whose outcomes completely determine syntactic structure. The set of derivations on a PCFG fits this bill. A potentially infinite but nonetheless discrete set, its members encapsulate everything there is to know about the syntactic structures the PCFG defines. Denoting the grammar at hand by $G$, let $TREE_G$ be a random variable whose values are derivations on $G$ and let $W$ be a string-valued random variable whose outcomes are in the language of $G$. Then the information conveyed by the first $i$ words of a sentence generated by $G$ is

$I(TREE_G|W_{0\cdots i}) = H(TREE_G) - H(TREE_G|W_{0\cdots i})$ (2.2)

As standardly defined (Cover and Thomas, 1991), information conveyed is the reduction in entropy of one random variable by discovering the outcome of another. To characterize the information conveyed to the human sentence processor, consider the information $I$ conveyed by just the $i^{th}$ word $w_i$ given the preceding words. Equation 2.3 expresses this quantity as a difference in the conditional entropy of $TREE_G$.

$I(TREE_G|W_{0\cdots i}) - I(TREE_G|W_{0\cdots i-1}) = [H(TREE_G) - H(TREE_G|W_{0\cdots i})]$

$- [H(TREE_G) - H(TREE_G|W_{0\cdots i-1})]$

$I(TREE_G|W_i = w_i) = -H(TREE_G|w_{0\cdots i}) + H(TREE_G|w_{0\cdots i-1})$

$= H(TREE_G|w_{0\cdots i-1}) - H(TREE_G|w_{0\cdots i})$ (2.3)
Equation 2.3 says that the answer to the question “how much information does a comprehender get from a word?” is the same as the answer to “how much was uncertainty about the derivation reduced?” The answer to this latter question is found in the work of Ulf Grenander (1967) which shows how to calculate the entropy of all derivations rooted in a particular nonterminal symbol. Grenander’s theorem 4.2 expresses the entropy of a grammar symbol as the sum of two quantities:

- the entropy of the choice of rule used to rewrite the symbol
- the expected entropy of any children.

These are the two terms of equation 2.5 whose interpretation is now considered more closely.

Continuing to symbolize as $G$ some given PCFG, let the set of production rules in $G$ be $\Pi$. For a given nonterminal $\xi$ the finite set of rules rewriting $\xi$ is denoted $\Pi(\xi)$. Define lowercase $h$ to be a vector indexed by nonterminal symbols. Each component, given in equation 2.4, contains the entropy of a single rewrite decision, whose outcomes are rule choices $r$ from $\Pi(\xi)$ having probability $p_r$.

$$h_i = h(\xi_i) = - \sum_{r \in \Pi(\xi_i)} p_r \log_2 p_r$$

(2.4)

Grenander found a recursion relation for the entropy of nonterminals in terms of $h$ and the expected entropy of the resulting children. Equation 2.5 depicts\(^1\) the general situation in which rule $r$ rewrites a nonterminal $\xi_i$ as $n$ daughters, $\xi_{j_1}, \xi_{j_2}, \ldots, \xi_{j_n}$.

$$H(\xi_i) = h(\xi_i) + \sum_{r \in \Pi(\xi_i)} p_r [H(\xi_{j_1}) + H(\xi_{j_2}) + \cdots]$$

(2.5)

Letting $A$ be the expectation matrix of grammar $G$ (see Appendix) the solution to this recursion can be expressed succinctly as the matrix equation 2.6 where $I$ is now the identity matrix and $h$ is given component-wise by equation 2.4. $H_G$ is similarly given component-wise by equation 2.5.

$$H_G = h + AH_G$$

$$h = H_G - AH_G = (I - A)H_G$$

$$H_G = (I - A)^{-1}h$$

(2.6)

Consistency ensures that $(I - A)^{-1}$ exists. $H_G$ is then a vector indexed by the nonterminals

\(^1\)N.B. the definition of uppercase $H$, the entropy of a nonterminal, uses lowercase $h$, the entropy of the single-rewrite decision.
procedure $H(D_A)$
initialise the result $temp = 0$
partition $D_A$ into $k$ classes sharing the same rule $A \rightarrow \gamma$ weighted with probability $p$
divide the probabilities $p$ by their sum to obtain $k$ renormalized probabilities $p_{\text{norm}}$
for $i = 1$ to $k$ do
  add $p_{\text{norm}} \log_2 p_{\text{norm}}$ to $temp$
  let $n$ be the minimum number of expanded daughters of $A$ across derivation trees in class $i$
  let $\beta$ be the substring of $\gamma$ from the $n^{\text{th}}$ symbol to the end
  let $v$ be a vector indexed from 1 to $n$
  for $j = 1$ to $n$ do
    set $v_j$ to be the union of all derivation trees rooted in the $j^{\text{th}}$ nonterminal from class $i$
    add $p_{\text{norm}} [H(v_1) + H(v_2) + \cdots + H(v_n) + H_{\text{expected}}(\beta)]$ to $temp$ (* recur on big $H$ *)
  end for
end for
return $temp$

Figure 2.4: Computing the entropy of the partial derivations common to $D_A$

of $G$. Since all derivations begin with the start symbol $S$, $H_{G(S)} = H(TREE_G)$.

2.3.3 Conditional entropy

Grenander’s result shows how to compute $H_{G(X)}$ for all nonterminals $X$ in one
step by inverting a matrix. However, to calculate how much information a comprehender
gets from a word using equation 2.3, the conditional entropy $H(TREE_G|w_{0..i})$ is needed.
Grenander’s recursion relation is again applicable, but only to derivations resulting in the
left-prefix $w_{0..i}$. Counting up the entropy of just these derivations characterizes the uncer-
tainty of a parser state.

The algorithm in Figure 2.4 is designed to do this counting. Given a set $D_A$ of
weighted derivation trees all rooted in the same nonterminal, it recursively computes the
entropy of the tree set compatible with the prefix seen so far by applying equation 2.5. For
parse tree nodes dominating only unseen words, the entropy $H_{\text{expected}}$ is simply the entropy
of all grammatical possibilities, as specified by equation 2.6.

This chapter’s contribution is the observation that intermediate states of a top-
down parser are specifications of the classes of derivations that can derive the left-context,
followed by any continuation. This point has been made and applied quite productively
by Lang and colleagues (Lang, 1974; Lang, 1988; Billot and Lang, 1989). What has not
been observed is that the same methods for calculating the entropy of a nonterminal can
be applied to the grammars implicitly defined by intermediate parser states. This permits
the straightforward definition of ‘information conveyed by a word in a sentence’ which, I
suggest, is a quantity of some psycholinguistic interest.
2.3.4 Left recursion

There is one rather technical barrier to applying this idea. Even leaving some nonterminals unexpanded, the set of partial derivations generating some prefix of a sentence may not be finite. This is the problem faced by top-down parsers on left-recursive grammars: there is an infinity of possible analyses indexed by the number of cycles through applicable, left-recursive rule sets. A finite factorization of this infinity is needed so that the entropy of the distribution on derivations can be calculated\(^2\).

Rather than modifying the top-down parser, this problem can be dealt with by transforming the grammar to remove left recursion (Huang and Fu, 1971). Claims of strong equivalence are then justified to the extent that the transform is invertible.

2.4 Examples

2.4.1 Main-verb/reduced relative

The method described above provides a characterization of the famous garden path effect of Bever (1970) in terms of information conveyed. It suffices to consider only syntactic processing, since semantic rules are taken to be in one-to-one correspondence with syntactic rules (Steedman, 2000). Following Crain and Fodor (1985) and Gazdar et al. (1985), no further processor resources beyond those needed for parsing probabilistic context-free phrase structure grammar are assumed.

Grammar

Consider the grammar of reduced relative clauses shown in Figure 2.5. Key probabilities of this Initial Bever Grammar are set according to corpus frequencies compiled by Rohde (2002).

The morphology of ‘raced’ is ambiguous between the past participle and the simple past tense. Although lexical in origin, this ambiguity causes constructional ambiguity since it permits several possible phrase structures for the sentence “the horse raced past the barn.”

The grammar in Figure 2.5 expresses the traditional adjunction analysis of relative clauses with the left-recursive rule NP → NP RRC. This left recursion is removed in the grammar of Figure 2.6. In this weakly-equivalent but non-left-recursive grammar, the adjunction analysis is recoded as complementation in a way that can be seen in Figure 2.10, to be discussed later.

Applying equation 2.6, the entropy of each nonterminal is given in Figure 2.7.

\(^2\)An improved method of calculating this entropy from finite as well as infinite sets of derivations is presented in chapter 3.
Figure 2.5: Initial Bever Grammar

Figure 2.6: Non-left-recursive Bever Grammar
**nonterminal entropy**

<table>
<thead>
<tr>
<th>Nonterminal</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>5.07</td>
</tr>
<tr>
<td>PP</td>
<td>2.05</td>
</tr>
<tr>
<td>RRC</td>
<td>3.05</td>
</tr>
<tr>
<td>VP</td>
<td>3.02</td>
</tr>
<tr>
<td>DT</td>
<td>0.00</td>
</tr>
<tr>
<td>NN</td>
<td>1.00</td>
</tr>
<tr>
<td>Vppart</td>
<td>1.00</td>
</tr>
<tr>
<td>Vpast</td>
<td>1.00</td>
</tr>
<tr>
<td>IN</td>
<td>0.00</td>
</tr>
<tr>
<td>NP</td>
<td>2.05</td>
</tr>
<tr>
<td>Z0</td>
<td>4.09</td>
</tr>
</tbody>
</table>

Figure 2.7: Entropies of nonterminals in Non-left-recursive Bever Grammar

**States of a top-down parser**

The parser begins analyzing “the horse raced past the barn fell” having heard no words at all. At this point, the conditional entropy of the parser state given the (nonexistent) input string is just the entropy of the grammar \( H_G(S) = 5.07 \) bits.

Upon hearing the first word “the” two classes of analyses come into play, corresponding to the two ways the noun phrase could have been expanded. In Figure 2.8 and subsequent depictions, nonterminals with asterisked names are unexpanded.

![Figure 2.8: Parser state having heard “the”](image)

Considering only these classes of analyses, the conditional entropy of the start symbol S turns out to be equal to its unconditional entropy — because every sentence on this grammar starts with “the”, the word conveys no information at all. However, the next word, “horse,” is informative. As shown in Figure 2.9 the word “horse” following “the” conveys one bit because the speaker has chosen one of two equally-probable alternative ways of expanding NN.

![Figure 2.9: Parser state having heard “the horse”](image)
At “the horse raced” the parser state comprises four analyses, shown in Figure 2.10.

![Diagram of parser state](image)

Figure 2.10: Parser state having heard “the horse raced”

At the word “raced,” the parser first explicitly represents the distinction between main verb and reduced relative structures.

The rest of the words are processed similarly. For words that reduce entropy, the magnitude of the reduction is written in Figure 2.11. At “fell”, where the number of compatible derivations is reduced to one (from six), nearly four bits are conveyed—approximately 75% of the information specified by the grammar. This is the sense in which the garden path effect is characterized as confusion brought on by the sheer volume of information being processed. In rejecting the distinction between most-highly valued, within-beam, and out-of-beam analyses (Frazier and Clifton, 1996), this account requires relatively few assumptions compared to alternatives that invoke particular structure-building operations, beam-width constants or semantic contexts.

2.4.2 The NP/S and NP/Z ambiguities

Since the sort of predictions afforded by an information-based account are numerical, they can also be used to characterize processing asymmetries between sentences as well as within sentences.

The NP/S and NP/Z temporary ambiguities, illustrated below in (1) are asymmetric in just this way.

---

3 Section 2.4 uses the main verb/reduced-relative ambiguity to propose that the magnitude of the entropy reduction at disambiguating points is proportional to subjects’ comprehension effort generally. It is true, however, that the grammar of Figure 2.6 would predict high difficulty at a sentence-ending period if the S rule were changed to rewrite as NP VP PERIOD. This is because, structurally, the end-of-sentence signal is just as informative as the verb “fell.” Although a determination that visually-presented sentence-final period symbols in fact have this effect on human readers would be an interesting development, it is coincidental that the disambiguating region in this example falls at the end of the sentence. The larger mystery is why some disambiguations which seem to be implicated by linguistic grammars actually give the human processor so little trouble. Are these ambiguities artifacts of the deductive nature of linguistic theory? Or deep facts about the processor? Since the burden of section 2.4 is to demonstrate the converse—that some cases of effortful disambiguation can be modeled—it has little to say about this challenging puzzle.
<table>
<thead>
<tr>
<th>word</th>
<th>reduction in entropy (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0</td>
</tr>
<tr>
<td>horse</td>
<td>1</td>
</tr>
<tr>
<td>raced</td>
<td>0.123</td>
</tr>
<tr>
<td>past</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>0</td>
</tr>
<tr>
<td>barn</td>
<td>0.123</td>
</tr>
<tr>
<td>fell</td>
<td>3.82</td>
</tr>
</tbody>
</table>

Figure 2.11: Bits of entropy reduction for a garden path sentence

(1) a. The Australian woman saw the famous doctor had been drinking quite a lot.

b. Before the woman visited the famous doctor had been drinking quite a lot.

The NP/S ambiguity comes about because the verb “saw” can either take an SBAR or an NP complement; the human sentence processing mechanism is lured up the garden path by the NP complement analysis until it reaches the auxiliary “had”, where it becomes apparent that only the SBAR subcategorization can be correct. Likewise, the NP/Z ambiguity is due to the verb “visit” taking either zero or one NP complements.

Sturt, Pickering, and Crocker (1999) found that the garden path effect for the NP/Z ambiguity (1b) was more than four and a half times the size of the effect in the NP/S ambiguity (1a). This asymmetry has been interpreted in terms of the Theta Reanalysis Constraint of Pritchett (1988) and others. Sturt and Crocker suggest that the destructive nature of the reanalysis required in the NP/Z but not the NP/S ambiguity can
also explain the asymmetry\(^4\). Yet another alternative is that the magnitude of information transacted at the disambiguating point is different across the two sentences.

To illustrate this alternative, consider the grammar in Figure 2.12.

The grammar in Figure 2.12 generates both of the sentences given in (1). At the disambiguating word “had” in (1a), about 3.45 bits are transacted. In (1b) at the same disambiguating word “had” about 8.79 bits are transacted. This demonstrates that the reading time asymmetry between the NP/S and NP/Z cases can be modeled as an information

\(^4\)Pritchett (1988) derives the Theta Reanalysis Constraint from the theta theory module of Government-Binding (GB) theory (Chomsky, 1981). Theta theory is a kind of dependency grammar overlayed on top of phrase structure that ascribes ‘thematic’ roles like agent, patient, experiencer etc. to the arguments of verbs. That a noun phrase stands in one of these relationships with a verb is, in GB, called ‘theta-marking.’ Pritchett’s constraint is repeated here as 1.

(1) Theta Reanalysis Constraint: Syntactic reanalysis which interprets a \(\theta\)-marked constituent as outside its current \(\theta\)-domain is costly.

\(\theta\)-domain: \(\alpha\) is in the \(\theta\)-domain of \(\beta\) iff \(\alpha\) receives the \(\gamma\) \(\theta\)-role from \(\beta\) or \(\alpha\) is dominated by a constituent that receives the \(\gamma\) \(\theta\)-role from \(\beta\).

Pritchett suggests that in a sentence like 2

(2) I warned her mother loathed me.

the theta-role goal is initially assigned to the pronoun \(\textit{her}\), which is later expanded unproblematically to the whole NP \(\textit{her mother}\). Then, when the lower verb \(\textit{loath}\) is encountered, syntactic reanalysis occurs such that \(\textit{her mother}\) leaves the goal domain to be reinterpreted as part of a proposition domain, violating the Theta Reanalysis Constraint. This is in contrast to 3

(3) We know her contributions failed to come in.

where, although “\(\textit{her contributions}\) is reinterpreted as the \(\textit{agent}\) of \(\textit{fail}\)... it remains within the \(\textit{patient}\) \(\theta\)-domain of \(\textit{known}\) as it is the subject of a clause which itself receives the \(\textit{patient}\) \(\theta\)-role” (page 545). Sentence 3 is analogous to sentence 1a and 2 is analogous to sentence 1b.

Sturt, Pickering, and Crocker (1999) offer a similar proposal. In the spirit of Marcus, Hindle, and Flec (1983) they suggest that in 1a, reanalysis preserves a domination relation between the VP saw the famous doctor and the NP the famous doctor. By contrast, in 1b the VP visited the famous doctor dominates the NP the famous doctor only up until the disambiguation point. After reanalysis, these nodes do not stand in a domination relationship. In this sense, parsing the NP/S ambiguity is monotonic in a way that the NP/Z ambiguity is not.
<table>
<thead>
<tr>
<th>Probability</th>
<th>Symbol</th>
<th>→</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>S</td>
<td>→</td>
<td>NP VP</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>S</td>
<td>→</td>
<td>PP SBAR</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>SBAR</td>
<td>→</td>
<td>NP VP</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>NP</td>
<td>→</td>
<td>SPECNP NBAR</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>SPECNP</td>
<td>→</td>
<td>DT</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>NBAR</td>
<td>→</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>VP</td>
<td>→</td>
<td>V[SUBCAT2] NP</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>VP</td>
<td>→</td>
<td>V[SUBCAT1]</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>VP</td>
<td>→</td>
<td>V[SUBCAT3] SBAR</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>VP</td>
<td>→</td>
<td>V[SUBCAT4,ASP] VBAR[PRP,COP]</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>VBAR[PRP,COP]</td>
<td>→</td>
<td>V[SUBCAT4,ASP] VBAR[PRP,COP] VBAR[PRP]</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>VBAR[PRP]</td>
<td>→</td>
<td>V[SUBCAT2,PRP] ADVP</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>PP</td>
<td>→</td>
<td>PBAR SBAR</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>PBAR</td>
<td>→</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>P</td>
<td>→</td>
<td>before</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>ADVP</td>
<td>→</td>
<td>quite a lot</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>V[SUBCAT4,ASP]</td>
<td>→</td>
<td>had</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>V[SUBCAT7,PRP,COP]</td>
<td>→</td>
<td>been</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>V[SUBCAT2,PRP]</td>
<td>→</td>
<td>drinking</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>V[SUBCAT2]</td>
<td>→</td>
<td>visited</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>V[SUBCAT2]</td>
<td>→</td>
<td>saw</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>V[SUBCAT1]</td>
<td>→</td>
<td>visited</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>V[SUBCAT5]</td>
<td>→</td>
<td>saw</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>DT</td>
<td>→</td>
<td>the</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>ADJ</td>
<td>→</td>
<td>famous</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>ADJ</td>
<td>→</td>
<td>Australian</td>
<td></td>
</tr>
<tr>
<td>0.33</td>
<td>N</td>
<td>→</td>
<td>ADJ N</td>
<td></td>
</tr>
<tr>
<td>0.33</td>
<td>N</td>
<td>→</td>
<td>woman</td>
<td></td>
</tr>
<tr>
<td>0.33</td>
<td>N</td>
<td>→</td>
<td>doctor</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.12: Grammar for NP/S and NP/Z ambiguities
processing asymmetry\textsuperscript{5}.

2.4.3 Subject and Object relatives

The contrasting difficulty presented by subject and object relative clauses is an established processing asymmetry (see references in Gibson (1998)). From this point on, a single GPSG-type grammar, shown in Figure 2.13, will be adopted.

Taking reading time to be proportional to entropy reduction, this grammar derives the subject/object asymmetry – at least in rough outline. A reading time peak is predicted on the main verb, along with a smaller peak on the embedded verb of just the object relative.

The graphs in Figures 2.14 and 2.15 show the result of a regression of the predictions derived using the grammar of Figure 2.13 to the average regional reading times measured by Grodner, Watson, and Gibson (2000). The correlation coefficient is 0.49, \( p < 0.01 \).

At the embedded verb in the object- (but not the subject-) relative, entropy is reduced by about 10 bits. This happens in the object relative (Figure 2.15) because the comprehender can determine at “sent” that there will be no recursive modification of the noun phrase “the photographer.” By contrast, in the subject relative (Figure 2.14), modification of the initial noun phrase “the reporter” has already been signaled by “who,” a word which opens up at least as many possibilities as it resolves. Since there is no overt noun phrase available for modification, the verb does not serve to disconfirm this possibility, and entropy is not reduced in the same way. The more general claim, if the grammar in Figure 2.13 is accurate, is that subject relative clauses are read more quickly on the embedded verb because the human sentence processing mechanism does not have to cope with the possibility, at that

\textsuperscript{5}As pointed out to me by William Badecker, this grammar does not literally cover the unambiguous versions of Sturt et al.’s (1999) stimuli. To cover all four classes of stimuli, two additions to the grammar of figure 2.12 are necessary. The first change is that an overt complementizer needs to be optionally generated. This can be accomplished by adding rules such as

\[
\begin{align*}
\text{SBAR} & \rightarrow \text{C} \ \text{NP} \ \text{VP} \\
\text{C} & \rightarrow \text{that}
\end{align*}
\]

Secondly, an optional comma needs to be available to constrain \textit{visited} to be in subcategorization class 1. To do this, add the following rules:

\[
\begin{align*}
\text{S} & \rightarrow \text{PP} \ \text{COMMA} \ \text{S} \\
\text{COMMA} & \rightarrow ,
\end{align*}
\]

These rules yield similar predictions on the ambiguous stimuli, but (spuriously) predict that the ambiguous version of 1a is easier than unambiguous control. This is because the uncertainty of the parser state is brought down by the possibility that the sentence could be over before the disambiguating word \textit{had} in 1a. Paralleling the discussion in footnote 3, it should be stressed that the goal of subsection 2.4.2 is the presentation of an account of the processing difficulty of the ambiguous stimuli.
Figure 2.13: PCFG for center-embedding and Grodner et al. (2000) subject-object stimuli
The reporter who sent the photographer to the editor hoped for a good story.

**Figure 2.14:** Subject relative (observed data from Grodner et al. (2000))

The reporter who the photographer sent to the editor hoped for a good story.

**Figure 2.15:** Object relative (observed data from Grodner et al. (2000))
level of embedding  total entropy reduction
0        20.52
1        38.37
2        47.08

Figure 2.16: Total information transacted at increasing levels of center-embedding point, of recursive modification.

2.4.4 Center-embedding

It has been known at least since Yngve (1960) that center-embedding induces processing difficulty, quite apart from any apparent ambiguity. Following Gibson (1998; 2000) observe that the sentences in (2) are increasingly center-embedded

(2)  a. The reporter disliked the editor.
    b. The reporter \([s'] who the senator attacked \] disliked the editor.
    c. The reporter \([s'] who the senator \([s' who John met \] attacked \] disliked the editor.

Using the same GPSG-type grammar (Figure 2.13), the increasing processing difficulty associated with center-embedded sentences can be modeled by the increasing total amount of information conveyed. For the sentences in (2), these values are tabulated in Figure 2.16.

While 47 bits are transacted processing a sentence like (2c), only 24 bits are needed for a right-branching sentence of the same length such as example (3).

(3) John met the senator who attacked the reporter who disliked the editor.

This demonstrates that the correctness of the prediction is not trivially due to the increasing length, but rather to the increasing level of embedding of the sentences involved. In fact, the final three verbs in the center-embedded sentence (2c) convey much more structural information than in their right-branching counterpart. This is because each verb resolves a structural decision between completion and continued modification of a noun phrase, a disambiguating role they do not play in the right-branching control.

2.5 Conclusion

The burden of the demonstrations in the last section was not to provide a tighter fit to the empirical data than was previously available. Rather, the demonstrations are meant to suggest that a range of cognitive load phenomena that have been extensively studied
in sentence processing may have an intriguing explanation at a fundamental level; a level whose existence follows from viewing parsing as a computation in which the ambiguity of a grammar is systematically reduced as more words are processed. The proposal is that human readers are able to identify combinatorial relationships in sentences because at each word they are performing the maximum amount of disambiguation, an amount of work proportional to the information conveyed by the word.

To compute the information conveyed, the procedure in Figure 2.4 was used in conjunction with a symbolic, top-down parser. The operation of this mechanism is not proposed as a cognitive process; it only serves to calculate the consequences of the theoretical claim. In fact, the issue of what cognitively- (or neurally-) plausible processing architectures actually reduce entropy as specified here is very much open. It may be that the entropy reduction for individual parser actions can be calculated and used in a serial parser. Such a model might operate by greedily reducing as much entropy as possible, backtracking when blocked. However, the work surveyed at the beginning of the chapter suggests that an array of diverse connectionist alternatives is also possible. If Harmony Theory (Smolensky, 1985) can be applied to one or another of these alternatives, it ought to be possible to derive entropy-reducing probabilistic processing from the Harmony values of the analyses considered within a parser state. To the extent that these values reflect substantive linguistic properties, a connectionist ‘constraint-based’ approach may offer the hope of breaking the endless circle of appealing to frequency as an explanation for performance.

2.6 Appendix: Consistency of PCFGs

It was observed early on by Grenander (1967) and Ellis (1969) that normalizing the rules of a probabilistic context-free grammar is insufficient to ensure that the defined probability model is a proper one. The probability model defined by a PCFG is consistent if the probability assigned to all sentences in the language of the grammar sums to 1.0.

Consider the normalized rules in Figure 2.17. Taking S as the start symbol, and lowercase ‘a’ as the only terminal, this grammar generates the language $a^n$. However, the probability assignment to the space of strings of a’s only adds up to $\frac{1}{2}$ (this is shown generally in Ellis (1969) and Grenander (1967)). Intuitively, the reason is that it’s more likely that the self-duplicating S rule will be selected, rather than the derivation-ending preterminal S rule. So some derivations, the infinite ones, go on forever, sapping away probability mass from those that do end. The grammar is inconsistent.

A fully satisfactory consistency condition for PCFGs relies on a view of the derivation process as a branching process (Harris, 1963). Its statement requires the concept of an expectation matrix. The expectation matrix $A$ of a grammar is a square matrix indexed
\[ S \rightarrow S \ S \]
\[ S \rightarrow a \]

Figure 2.17: Inconsistent PCFG

by nonterminal-symbols where each entry \( a_{ij} \) contains the sum of probabilities that the \( i^{th} \) symbol is rewritten as the \( j^{th} \) symbol. \( A \) represents the ‘fertility’ of each grammar symbol as regards each possible kind of daughter. A grammar is consistent if each kind of child is destined to eventually die out – that is, be rewritten by only terminal symbols. This property holds if the largest eigenvalue (the ‘spectral radius’) of the grammar’s expectation matrix is less than 1. Although calculation of the dying-out probability is a standard topic in stochastic processes (see for example section XII.5 of Feller (1957) or section 6.7 of Grimmett and Stirzaker (1992)) it was Grenander (1967) who first saw the application to probabilistic grammars. Section 3.10 discusses a way to bring inconsistent PCFGs back to consistency without altering the linguistic analyses they express.
Chapter 3

A probabilistic, mildly-context-sensitive hypothesis about human sentence processing

...it might be useful to look into the properties of a perceptual model $M$ with two basic components, $M_1$ and $M_2$ operating as follows: $M_1$ contains a small, short-term memory. It performs computations on an input string $x$ as it is received symbol by symbol and transmits the result of these computations to $M_2$. $M_2$ contains a large long-term memory in which is stored a generative grammar $G$; the task of $M_2$ is to determine the deeper structure of the input string $x$, using as its information the output transmitted to it by $M_1$.

... It might be useful, therefore, to develop measures of various sorts to be correlated with understandability.

(Miller and Chomsky, 1963, 476,480)

[our] chapter has demonstrated the role of syntactic analysis in the processes of reading English text. It has long been known that a person’s expectancies play a major role in his perceptual processes. In this chapter, we describe our pioneering efforts to model the dynamic nature of the change in expectancies that occurs as the reader progresses through the words of a text.

(Stevens and Rumelhart, 1975, 155)

3.1 Orientation: which field?

The present chapter is intended as a contribution to a venerable research program in cognitive science exemplified by the passages quoted above. This research program amounts to a set of persistent attempts to formulate explicit, general ideas about people’s ability to read natural languages like English. To that end, section 3.2 formalizes a particular version of what Stevens and Rumelhart call change in expectancies; it follows Suppes
(1970) in the adoption of probabilistic grammars. This leading idea will be the heart of a proposal about human sentence processing difficulty. However, a fully explicit model will require several other ingredients. Indeed I will go on to defend Chomsky’s early point that an adequate perceptual model requires sentence processing theorists to take up particular positions on linguistic structure. Section 3.4 consequently focuses on what Chomsky and Miller call $M_2$ in explaining how a particular kind of generative grammar due to Stabler (1997) can encode a reader’s knowledge of language. Grammars of this type fall into the class of ‘mildly context-sensitive grammars’ (Joshi, Vijay-Shanker, and Weir, 1991). This class is widely viewed as being expressive enough to accommodate reasonable structural descriptions of sentences in all the natural languages while still ruling out some conceivable artificial languages. A small grammar fragment for English written in this formalism (discussed in more detail in chapter 4) is used to derive predictions about the relative understandability of a hierarchy of sentence types. This hierarchy is of interest because it has been proposed as a linguistic universal notably by Keenan and Comrie (1977). These predictions are then correlated with the hierarchy as a psycholinguistic reality (Keenan and Hawkins, 1987) in section 3.3.

### 3.1.1 The Competence hypothesis as a methodological principle

Throughout, I will follow Chomsky (1965) in supposing that “...a reasonable model of language use will incorporate, as a basic component, the generative grammar that expresses the speaker-hearer’s knowledge of language” (page 9). This position has been dubbed the *competence hypothesis* by Bresnan (1982a). Stabler (1984) points out that to do otherwise would multiply the explanatory burden of accounting for the language-understanding behaviors people exhibit. If the human sentence processor “incorporates as a basic component” the kind of grammar linguists discover, this immediately raises the question of how rules of this grammar are used. A strong version of the competence hypothesis would hold that

The principles of the competence grammar are directly used by the human language processor in constructing a syntactic structure and interpreting it.

(Stabler, 1991, 199)

The Strong Competence hypothesis highlights the difficulty of adapting linguistic grammars to the problem of constructing semantic forms for incomplete segments of sentences. Although a variety of ingenious proposals have been offered (Abney, 1989; Steedman, 1989; Shieber and Johnson, 1993) there remains no clear consensus about what meanings of initial segments of natural language sentences are. As a result it has become difficult to either confirm or disconfirm Strong Competence.
This work adopts the competence hypothesis in a weakened version. While still retaining the mentalistic concept of a grammar to “integrate and explain behavioral data” (Miller, 2003, 141), the more modest version claims only that the processor searches through the structures defined by the grammar in some way, on the assumption that any processing architecture will have to deal with the same fundamental kinds of uncertainty regarding the structure being perceived. Methodologically, this is the claim that more accurate competence grammars will lead to better sentence processing models.

Stepping back from particular algorithms in this way is meant as an acknowledgment that our ideas about linguistic structure are constantly developing, and that the relevance of evidence regarding linguistic structure – for instance judgments of acceptability, or entailment – depends crucially on a constellation of auxiliary assumptions about the form of grammar and its instantiation in the processor.

As a methodological principle, then, a minimum of extra-grammatical machinery will be assumed in this chapter. The objective will be the most accurate and insightful picture of the human sentence processor possible while keeping this constellation of additional assumptions as small as possible.

### 3.1.2 Probabilistic grammar

Before firmly resolving to follow this methodology faithfully for the rest of the chapter, one addition to the traditional conception of a generative grammar seems unavoidable: frequency. The differential processing characteristics of highly frequent versus infrequent stimuli have been documented extensively in psycholinguistics (MacDonald, Pearlmutter, and Seidenberg, 1994; Mitchell et al., 1995; Trueswell, 1996). Likewise, the utility of frequency information in computational linguistics can hardly be overstated\(^1\).

The assumption of probabilistic grammar goes against the general strategy laid out in section 3.1.1. This is true in the strict sense that probabilistic grammars define not only a set of generated sentences, but additionally impose a probability distribution on that set\(^2\).

While costly, making grammars probabilistic has the benefit of making even incomplete ideas from linguistics immediately applicable to questions of processing. By ‘complete’ I mean that no presently-existing grammar encodes all the factors that govern even putatively invariant properties like sentence acceptability. There will always be a residue of

---

1 A somewhat dated account of the statistical revolution in natural language processing can be read in volume 18, number 4 of AI Magazine. Probability in syntax and human syntactic processing are treated in the Manning and Jurafsky chapters, respectively of Bod, Hay, and Jannedy (2003).

2 As Suppes (1970) points out, the correctness of this additional prediction can be evaluated using the usual techniques from statistics.
factors that must be controlled for. Discourse context, dialectal status and relative attention of the judgment maker – to say nothing of lexical frequency or real-world plausibility – while often not the focus of attention in formal linguistics, can nevertheless decisively influence the shape of its empirical basis.

Probabilistic grammars alleviate this problem by admitting that our theories of natural language are incomplete. As in other domains where complete theorization is either impossible or impractical, they analyze some features of the domain as simply random. This allows inquiry to proceed with as much insight as is available, and synergizes two kinds of explanation that could potentially be complementary. In the clearest cases it may be possible to identify the probability model for a certain grammar as encoding only extra-grammatical facts, whereas the categorical part of the grammar might serve to encode all of the properly grammatical facts. However such a happy coincidence is by no means necessary and will not be assumed in this chapter.

3.2 The Entropy Reduction hypothesis

What will be assumed is

**Hypothesis 1 (Entropy Reduction Hypothesis – intuitive)** the time it takes for a person to read a word in a sentence is related to the number of bits signaled to the person by that word with respect to a probabilistic grammar the person knows.

Chapter 2 explored the consequences of this hypothesis if the grammar is a probabilistic context-free grammar (PCFG). The present chapter extends this exploration to a more expressive formalism.

The main idea\(^3\) is that uncertainty about the set of PCFG derivations compatible with the words seen so far fluctuates as new words come in. Fluctuations in this value are taken as a psycholinguistic prediction.

For instance, the very simple PCFG\(^4\) in figure 3.1 generates two sentences. There are exactly two derivations, each with probability one-half.

\[
\begin{align*}
0.5 & \quad S \rightarrow \text{john loves mary} \\
0.5 & \quad S \rightarrow \text{john sleeps}
\end{align*}
\]

Figure 3.1: A very simple PCFG

\(^3\)A similar idea was independently pursued earlier in the more restricted setting of “reduced” deterministic finite automata by Grassberger (1986).

\(^4\)The convention that the start symbol is $S$, that nonterminals are written in capital letters and that terminals are in lowercase will be adhered to throughout.
Upon hearing the first word of a sentence in the language of this grammar, no information is conveyed to the hearer. Since all derivations are compatible with the prefix string “john...” it is only the second word that eliminates one of the derivations, conveying a single bit. Said another way, finding out that the second word is “loves” rather than “sleeps” reduces the entropy of the start symbol S from 1 bit to 0 bits. On this view, nonterminals like S are random variables that have as outcomes the right-hand sides of rules that rewrite them.

In figure 3.1 it is easy to see what the set of alternative derivations is; in principle its members could be written out. Such enumeration is not effectively possible in a recursive grammar like the one in figure 3.2.

\[
\begin{align*}
0.5 & \quad S \rightarrow \text{john thinks CP} \\
0.5 & \quad S \rightarrow \text{john sleeps} \\
1.0 & \quad \text{CP} \rightarrow \text{that S}
\end{align*}
\]

Figure 3.2: Recursive PCFG

The grammar in figure 3.2 defines an infinite number of derivations, whose probability tails off with their size so that the sum of the probabilities assigned to the language is exactly 1.0.

<table>
<thead>
<tr>
<th>word #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>\cdots</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>john</td>
<td>sleeps</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>john</td>
<td>thinks</td>
<td>that</td>
<td>john</td>
<td>sleeps</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>john</td>
<td>thinks</td>
<td>that</td>
<td>john</td>
<td>thinks</td>
<td>that</td>
<td>john</td>
<td>sleeps</td>
<td></td>
</tr>
</tbody>
</table>

\vdots

Figure 3.3: Sentences generated by recursive PCFG

On this grammar two bits are conveyed if “sleeps” is the second – or fifth, eight, eleventh, etc – word. Even though there are an infinite number of derivations licensed by this grammar, it is possible to compute the entropy of the start symbol by applying a theorem due to Ulf Grenander (1967).

**3.2.1 Entropy of nonterminals in a PCFG**

Grenander’s theorem specifies a recurrence relation that gives the entropy of each nonterminal in a PCFG G as the sum of two terms. Let the set of production rules in G be \( \Pi \) and the subset rewriting nonterminal \( \xi \) be \( \Pi(\xi) \). Denote by \( p_r \) the probability of a rule \( r \). Then
\[ h_i = h(\xi_i) = - \sum_{r \in \Pi(\xi_i)} p_r \log_2 p_r \]

\[ H(\xi_i) = h(\xi_i) + \sum_{r \in \Pi(\xi_i)} p_r [H(\xi_{j_1}) + H(\xi_{j_2}) + \cdots] \]

(Grenander, 1967, 19)

the first term, lowercase \( h \), is simply the definition of entropy for a discrete random variable. The second term, uppercase \( H \), is the recurrence. It expresses the intuition that derivational uncertainty is propagated from children to parents. For PCFGs that define a probability distribution (see section 2.6) the solution to this recurrence can be written as a matrix equation where \( I \) is the identity matrix, \( \vec{h} \) the vector of the \( h_i \) and \( A \) is a matrix whose \((i, j)\)th component gives the expected number of nonterminals of type \( j \) resulting from nonterminals of type \( i \).

\[ H = (I - A)^{-1} \vec{h} \quad (3.1) \]

(Grenander, 1967, 19)

### 3.2.2 Incomplete sentences

Grenander’s theorem supplies the entropy for any PCFG nonterminal in one step by inverting a matrix. To determine the contribution of a particular word, one would like to be able to look at the change in uncertainty about compatible derivations as a given prefix string is lengthened. When this set, the set of derivations generating a given string \( w = u_0w_1 \ldots w_n \) as a left prefix, is finite, it can be expressed as a list. In the case of a recursive grammar such as the one in figure 3.2 this set is not finite and some other representation is necessary.

Bernard Lang and Sylvie Billot (Lang, 1974; Lang, 1988; Billot and Lang, 1989) observed that this other representation can be another, related grammar. Billot and Lang showed how parsing could be viewed as intersection of a grammar with a regular language, of which ordinary input strings are but the simplest examples. This perspective readily accommodates the view of incomplete strings as regular languages whose members all have the same initial \( n \) words but continue with all possible words of the terminal vocabulary, for all possible lengths.
Chart parser states as grammars

The status of context-free grammars as the result of intersecting a regular input with a given context-free grammar can be appreciated by looking at chart parsing (Kay, 1986). The chart in chart parsing is typically a two-dimensional array, whose cells record sets of nonterminals in some way or another. As parsing happens, the array becomes populated with information about the presence of constituents e.g. that there is a noun phrase (NP) spanning positions 2 through 5. This would be indicated by adding NP to the set of nonterminals in the (2,5) cell of the chart. In a recognizer, all that matters is the presence or non-presence of the start symbol in the cell whose left and right positions correspond to the beginning and end of the input sentence. In a parser, derivations must be recovered. This is typically done by augmenting chart cells with back pointers that record the addresses of any child constituents. The key point is that a pair of back pointers picking out, for instance, Determiner in cell (2,3) and Noun in cell (3,5) define a kind of grammar rule

$$NP_{(2,5)} \rightarrow D_{(2,3)} \ N_{(3,5)}$$

Figure 3.4: “Situated” grammar rule

This kind of grammar rule refers not just to nonterminals in general, but to nonterminals situated in the string being parsed. Crucially, the chart keeps only single entries for nonterminals participating in derivational loops. For instance, if a noun phrase can be a bare noun and a bare noun can be a noun phrase a chart parser, having found a NP_{(2,5)} would also insert N_{(2,5)} in the same cell. Back pointers would be set up both from NP to N and from N to NP. In this way, back pointers record the possibility of recursion without falling into an infinite loop evaluating such recursion.

3.2.3 A more specific hypothesis

In light of Lang and Billot, the set of grammatical derivations of a string starting with $$w_0w_1 \ldots w_n$$ is well-defined and efficiently computable. It is intensionally represented by a grammar whose symbols are annotated with position indices as in figure 3.4. This grammar can be taken as the categorical part of a related probabilistic grammar whose entropy can be calculated using Grenander’s theorem\(^5\). A more specific version of hypothesis 1 can now be stated.

\(^5\)These related grammars can introduce uncongenial properties that can be overcome using a method described in the Appendix, section 3.10.
Hypothesis 1 (Entropy Reduction Hypothesis – precise) A person’s reading time at a word in a sentence is linearly related to any downward change in the entropy of the set of derivations generating the observed words as a prefix.

Standardly, whenever entropy is reduced it is said that information has been conveyed. The “expectancies” of Stevens and Rumelhart (1975) have taken the form of the chart-grammar encoding all possible grammatical derivations that continue the observed string. These expectancies are weighted, if not “integrated and explained,” by the probabilities of a probabilistic grammar, and it is a collective property – the change in their entropy – that is proposed as a psycholinguistic model.

With the tools of subsections 3.2.1 and 3.2.2, the predictions of hypothesis 1 are deducible from a probabilistic grammar. Section 3.4 describes a way this grammatical component of the psycholinguistic hypothesis can naturally include a variety of ideas from the syntax literature. But first, a discussion of the empirical domain at issue will be undertaken in section 3.3.

3.3 The Accessibility Hierarchy

The empirical domain is that of noun phrases containing relative clauses. Relative clauses constitute a fascinating puzzle for sentence processing theories at least in part because their non-canonical word order suggests a kind of deeper, more grammatically-oriented kind of processing. Much work (Bever, 1970; Wanner and Maratsos, 1978; Gibson, 1998) has focused on these constructions, primarily on relativization of the grammatical relations Subject and Object. These are two points on a scale known as the Accessibility Hierarchy.

The Accessibility Hierarchy (AH) is a cross-linguistic generalization about relative clause formation in natural languages discovered by Keenan and Comrie (1977). The generalization is an implicational markedness hierarchy of grammatical relations⁶ that can be ‘relativized’. The presupposition is that relativization is a rule (or rule schema) of all natural languages.

SUBJECT ⊲ DIRECT OBJECT ⊲ INDIRECT OBJECT ⊲ OBLIQUE ⊲ GENITIVE ⊲ OCOMP

Figure 3.5: The Accessibility Hierarchy of relativizable grammatical relations

⁶Keenan and Comrie (1977) define specific criteria for each grammatical relation. Oblique NP are typically preceded by prepositions in English, Genitive suffixed with ’s, and Objects of Comparison (OCOMP) preceded by than.
This hierarchy (figure 3.5) shows up in a variety of modern syntactic theories that have been influenced by Relational Grammar (Perlmutter and Postal, 1974). In Head-driven Phrase Structure Grammar (Pollard and Sag, 1994) the hierarchy corresponds to the order of elements on the SUBCAT list, and interacts with other principles in explanations of binding facts. The hierarchy also figures in Lexical-Functional Grammar (Bresnan, 1982b) where it is known as Syntactic Rank.

On the basis of about 50 languages, Keenan and Comrie found that if a language has a relative-clause formation rule applicable to grammatical relations at some point $x$ on the AH, then it can also form relative clauses on grammatical relations listed at all points before $x$.

### 3.3.1 The Keenan & S. Hawkins experiment

Keenan and Comrie (1977) speculated that their typological generalization might have a basis in performance factors. This idea was supported by the results of a psycholinguistic experiment done in 1974 that were not published until much later (Keenan and Hawkins, 1987)\(^7\).

Seeking some processing basis for the AH, Keenan and S. Hawkins conducted a study that examined people’s ability to comprehend, remember and produce sentences involving relative clauses from various points on the AH. They constructed four stimulus sentences exhibiting relativization on each grammatical relation\(^8\) using relatively frequent words – at least 50 per million in materials sampled by Thorndike and Lorge (1968) in the 1930s.

Both adults and children participated in the experiment; all were native British English speakers. Subjects heard each stimulus sentence read out loud on a tape. Half a second after the last word, adult subjects heard the names of eight digits. They were then expected to write down this digit sequence. Having completed the digit-memory interference task, subjects were finally asked to demonstrate their memory for the original stimulus sentence by writing it. These responses were coded for accuracy according to a point scheme.

Responses were graded on a 2,1,0 basis with 2 being the best and 0 the worst. A grade of 2 was assigned for essentially perfect repetition, allowing only a grammatically permissible change in relative pronoun (e.g., ‘that’ for ‘which’) and a grammatically permissible deletion of a relative pronoun.

\(^7\)A corpus study was also conducted whose results (also reported in Keenan (1987)) will be discussed and applied in section 3.6.

\(^8\)Consideration will be restricted here to the most well-established part of the AH. Keenan and Hawkins (1987) also considered Object of Comparison, and genitive as well as non-genitive Subjects of a Passive Predicate.
A grade of 0 was assigned if the response did not include a relative clause where the head has the same function as the stimulus.

In other cases errors were regarded as minor and the response assigned value 1. e.g tense change, verb particle omission, omission or incorrect addition of a noun modifier, lexical substitution with meaning retained, substitution of one proper noun for another, and incorrect recall of either the initial frame or the final transitive verb phrase.

(Keenan and Hawkins, 1987)

Under this coding scheme, response accuracy drops as the grammatical relation relativized-upon becomes more rare in the world’s languages (figure 3.6). These scores are then summed across adult and child subjects; overall children followed the AH more closely than adults.\(^9\)

<table>
<thead>
<tr>
<th>Grammatical Relation:</th>
<th>SU</th>
<th>DO</th>
<th>IO</th>
<th>OBL</th>
<th>GenS</th>
<th>GenO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repetition Accuracy:</td>
<td>406</td>
<td>364</td>
<td>342</td>
<td>279</td>
<td>167</td>
<td>171</td>
</tr>
<tr>
<td>errors (= R.A(_{\text{max}}) - R.A.)</td>
<td>234</td>
<td>276</td>
<td>298</td>
<td>361</td>
<td>471</td>
<td>469</td>
</tr>
</tbody>
</table>

Figure 3.6: results from Keenan & S. Hawkins (1987)

Keenan and Hawkins (1987) conclude that “the AH is reflected in the psychological processing of relative clauses in that repetition errors increase as the position of the relative clause on the hierarchy decreases.” However they are careful to say

It remains unexplained just why RCs should be more difficult to comprehend-produce as they are formed on positions lower on the AH.

(Keenan and Hawkins, 1987)

Hypothesis 1, if correct, would offer just such an explanation. If a person’s difficulty on each word of a sentence is related to the amount of derivational information signaled by that word, then the total difficulty reading a sentence ought to be the sum of the difficulty on each word. To propose otherwise would beg the question of which words subjects actually read and which they skip. The explanation would be that sentence understanders in general must do information-processing work on the scale of the entropy reduction brought about by each word, and that the sum of this work, on a per-sentence level increases with the AH.

3.3.2 A related proposal

In rough outline, this idea has already been proposed by J. Hawkins (1994). He suggests that the set of nodes in domination, sisterhood, or c-command relationships to a relativized structural position is the key to the AH. Defining these positions configurationally (Chomsky, 1965, 70-71), he observes that the cardinality of these sets increases

---

\(^9\)GenS and GenO stand for Genitive Subject and Genitive Object, respectively.
with the AH, and predicts that this complexity scale will correspond with ease of human processing.

This proposal follows Yngve (1960) and Miller and Chomsky (1963) in taking the number of parse tree nodes as a psycholinguistic prediction. The idea is plausible on the view of the human parser as “recovering” each node of the parse tree, an iterated step that takes a constant amount of cognitive work.

The sentence-level extension of hypothesis 1 refines J. Hawkins’ idea by supposing that larger parse trees are unproblematic for the human parser if their structure is largely determinate. It does not assume that finding an individual parse tree node is an atomic step of the human processing algorithm; rather it is the information-theoretic hardness of the problem the algorithm is solving that matters.

More practically, J. Hawkins assumes a parser that can recover parse trees in which fillers and gaps are co-indexed. The algorithmic recovery of such co-indexation has been a major research problem in computational linguistics since the beginning of transformational grammar (Joshi, 1968; King, 1983; Friedman, 1989; Johnson, 1991; Fong, 1991; Stabler, 1992). Since the cognitive issue is precisely how people are able to “unwind” such transformations, if they exist, a fully explicit proposal should specify exactly what grammatical relationships are encoded. This is the topic of section 3.4.

### 3.4 Minimalist Grammars

The competence hypothesis introduced as a methodological principle in subsection 3.1.1 implies that the grammatical assumptions inherent in a psycholinguistic model correspond directly with those made in linguistics. Any deviation from this general strategy renders a model less general in its dependence on parochial ideas about linguistic structure. At the same time, to be concrete, it is also desirable that the grammatical assumptions be formalized.

Minimalist Grammars (Stabler, 1997) (MGs) make explicit certain themes of Chomsky’s Minimalist Program (Chomsky, 1995a).

- words are bundles of features
- syntactic structure is created by rules named Merge and Move
- these rules respect a simple version of Chomsky’s “Shortest Move Condition”
- movement (as well as merger) always ‘checks’ features, meaning that checked features are removed from the computation
- ubiquitous empty categories are handled analogously to overt ones
The Minimalist Program is a research agenda within the transformational grammar tradition that is concerned with uncovering the respects in which human language is an optimal solution to the “design requirements” imposed by its relationships with other cognitive systems.

This controversial agenda (Johnson and Lappin, 1999) places a special emphasis on deriving previously discovered linguistic phenomena from abstract general principles. If successful, the Program might reconceptualize linguistics in the same way that “continuity” and “limit” rigorously reconceptualize calculus and the mathematics that uses calculus. Even if ultimately unsuccessful, the Program represents a vocabulary for discovery in which much current work in syntax is stated.

Stabler’s formalization is not only convenient for evaluating processing and other consequences of Minimalist proposals, but also addresses the long-standing problem in computational linguistics (Peters and Ritchie, 1973) of capturing transformational generalizations in an efficiently decidable theory. Other formalizations exist (Veenstra, 1998; Schneider, 1999) but their mathematical properties are less well-understood.

MGs have a movement rule, but the languages they generate are mildly context-sensitive. MGs are equal in weak generative capacity to set-local multi-component tree adjoining grammars (Michaelis, 1998). Viewed as a programming language for linguistics, MGs are well-suited for expressing linguists’ intuitions that “that moves here to check this, then this moves there to check that…”

### 3.4.1 A short tour of MGs

MGs generate trees of a certain kind by closing the functions **merge** and **move** on a lexicon of items possessing idiosyncratic features. The trees satisfy a headed $\overline{X}$ theory with only two bar levels: heads and phrases. Notation of these levels is customarily omitted and replaced with individual angle brackets indicating which binary subtree contains the phrase’s head.

$$
\begin{align*}
\text{XP} & \sim > \\
\overline{X'} & < \\
\overline{X} & \end{align*}
$$

Each of these trees has a certain string of *features* associated with it. Five feature types are possible.

$c, t, d, n, v, \text{pred}, \ldots$ category features

---

This section borrows expository material from a talk given by Henk Harkema at UCLA. Trees are drawn using Tk tools developed by Edward Stabler and collaborators.
=c, =t, =d, =n, =v, =pred, ...  selection features

+wh, +case, +focus, ...  licensor features

-wh, -case, -focus, ...  licensee features

Lavinia, Titus, praise, -s, ...  phonetic features

Lexical entries are trees of size zero and are associated with the longest feature strings. Such trees are called simple; all others are complex. Phonetic features of empty categories are empty strings (ε).

= n d - case every
= d = d v love
= t + wh c ε

...

The structure building functions operate on these trees and are defined case-by-case.

merge: If the left-most feature of the head of τ₁ is =x and the left-most feature of the head of τ₂ is x, then

\[
\text{merge}(τ₁, τ₂) = \begin{cases} < & \text{if } τ₁ \text{ is simple, and } \\ \frac{τ₁'}{τ₂} \end{cases}
\]

where τ₁' is like τ₁ except that feature =x is deleted, and τ₂' is like τ₂ except that feature x is deleted.

move: If the left-most feature of the head of τ is +y and τ has exactly one maximal subtree τ₀ the left-most feature of the head of which is -y, then

\[
\text{move}(τ) = \begin{cases} > & \text{if } τ \text{ is complex, and } \\ \frac{τ₀'}{τ'} \end{cases}
\]

where τ₀' is like τ₀ except that feature -y is deleted, and τ' is like τ except that feature +y is deleted and subtree τ₀ is replaced by a single node without features.
The Shortest Movement Constraint is construed as a mandate not to generate trees with two competing licensee features:

\[
\begin{array}{c}
\begin{array}{c}
< \\
+ f \beta \\
- f \gamma \\
- f \delta \\
\end{array}
\end{array}
\]

not in domain of \textit{move}

The \textit{closure} of a MG \( G \) is the set of trees generated by closing the lexicon under the structure building functions. A \textit{complete} tree is a tree which has only one syntactic feature (such as \( c \) for complementizer). This is the notion of start category for MGs. Finally, \( L(G) \) the language generated by \( G \), is the set of yields of complete trees in the closure of \( G \). There are many equivalent notations for MGs; a more formal presentation can be found in Stabler and Keenan (2003).

### 3.4.2 Context-free probability model

MGs are able to generate mildly context-sensitive languages because the \textit{move} rule is non-concatenative. A fundamental result, obtained independently by Harkema (2001) and Michaelis (2001) is that MGs are equivalent to Multiple Context-free Grammars (Seki et al., 1991) (MCFGs). MCFGs generalize standard context-free grammars\footnote{MCFGs also generalize the Linear Context-free Rewriting Systems of Weir (1988) by dropping some restrictions on string manipulation functions.} by allowing the string yields of daughter categories to be manipulated by a function other than simple concatenation. For instance, in a standard CFG, the yield of the parent category is always the string concatenation (\( \cdot \)) of the yields of the daughters.

\[
\begin{array}{lll}
\text{VP} & \rightarrow & \text{V} \quad \text{NP} \\
\text{s}^{-t} & \leftarrow & s \quad t \\
\text{S} & \rightarrow & \text{NP} \quad \text{VP} \\
\text{s}^{-t} & \leftarrow & s \quad t \\
\end{array}
\]

This means that if, say, a category deriving a WH-word appears in a verb phrase rule to the right of the verb, then any WH-words derived by that rule will also be to the right of the verb, as shown below.

\[
\begin{array}{lll}
\text{VP} & \rightarrow & \text{V} \quad \text{NP} \\
kiss \ who & \leftarrow & kiss \ who \\
\end{array}
\]
\[
S \rightarrow \text{NP \ VP}
\]

*John kiss who ← John kiss who*

In a non-concatenative MCFG, the yield of the daughter categories V and NP might be re-arranged in some other way.

\[
\begin{align*}
\text{VP} & \rightarrow \text{V \ NP} \\
(s, t) & \leftarrow s \ t \\
S & \rightarrow \text{NP \ VP} \\
&t \times r \times s & \leftarrow r \ (s, t)
\end{align*}
\]

For instance, the string re-arranging functions might transport the WH-word to the front of the sentence.

\[
\begin{align*}
\text{VP} & \rightarrow \text{V \ NP} \\
(kiss, who) & \leftarrow kiss \ who \\
S & \rightarrow \text{NP \ VP} \\
who \ John \ kiss & \leftarrow John (kiss, who)
\end{align*}
\]

The power of MCFGs derives from the ability of string handling functions (specified for each rule) to refer to \(n\)-tuples of categorized strings. This power is restricted by the finitude of these \(n\)-tuples.

While too involved to repeat in full detail here, the basis of Harkema and Michaelis’s result is the observation that the derivational possibilities for an MG tree structure are entirely determined by three factors: the syntactic features of the head, the tree’s status as being simple or complex, and the presence and type of other remaining licensee features in the tree. Coding all this information in a finite way defines the set of *relevant* categories. These are the classes of trees-associated-with-feature strings that could possibly be generated bottom-up by *merge* and *move*. Since MG lexicons are finite, there are a finite number of licensee feature types per grammar, and hence a finite tuple size required to keep track of them.

With the string manipulation functions fixed by the MG structure-building rules, it becomes possible to give a *derivation tree* for each sentence. Figure 3.7 shows a small MG, with the derivation tree for “the boy who I met” in figure 3.8 alongside the resulting structural description or ‘derived tree.’

As in Tree Adjoining Grammar (Joshi, Levy, and Takahashi, 1975), the nodes of derivation trees like the one on the left in figure 3.8 record instances of tree-combination. These derivation trees encode everything there is to know about an MG derivation, and
can be parsed in a variety of orders (Harkema, 2001). Most importantly, if equipped with weights on their branches, they can also be viewed as probabilistic context-free grammars. These weights can be set using any PCFG estimation method, and a specific technique will be described in section 3.6. Before turning to the probabilities, a linguistic application of the MG formalism will be discussed in section 3.5.

### 3.5 The Promotion Analysis

MGs are an attractive notation for encoding linguistic proposals because the primitive objects of the formalism—movement rules, features, tree positions—are the ones that figure in linguistic explanation. But the formalism itself is not a particular linguistic theory. Nor is it practical to consider all possible MGs\(^\text{11}\). This section presents an essentially linguistic argument that a certain pattern of English surface facts needs to be accounted for in the grammatical component \(M_2\) of a fully-specified psycholinguistic model. Then a particular analysis from the syntax literature is proposed and its concrete implementation as an MG is sketched. A grammar fragment using this analysis that covers the Keenan and S. Hawkins stimuli is discussed at length in chapter 4.

\(^\text{11}\)Even learning in the space of all possible context-free grammars is impractical, cf. “Why the simple approach fails” (Charniak, 1993, 7.1). Likewise, efforts to infer tree grammars from sentences always impose severe restrictions on possible trees (e.g., the restriction to a 96 tree inventory capable of deriving all possible binary bracketings in Schabes (1992)).
3.5.1 Relativization

The promotion analysis sketched in this section is an idea about the syntax of relative clauses. As a linguistic hypothesis, it grapples with agreed-upon English surface facts. The English surface facts are the ones that suggest the existence of long-distance dependencies between fillers and gaps in relative clauses, such as the sentences in 4.

(4)  a. the father explained the answer to the boy
    b. *the boy who the father explained the answer to the boy was honest
    c. the boy who the father explained the answer to was honest

Sentence 4a is perfectly acceptable on its own. But when embedded in the context “the boy who...was honest”, the result 4b is unacceptable. Acceptability can be restored by removing the inner copy of “the boy”, arriving at 4c. It is as if the sentence-initial noun phrase “the boy” depends on the non-presence of its subsequent repetition to guarantee the acceptability of the whole sentence.

Chomsky (1956) offered the kind of acceptability judgment pattern in 4 as a definition for the notion ‘dependency between positions in a sentence.’ In 4c the dependency is between the initial “the boy” and what must not occupy the position immediately after “to.” Since the number of intervening words between these two positions can be made arbitrarily large it is a long-distance dependency. These sorts of long-distance dependencies are uncontroversially implicated in human sentence processing (Stowe, 1985) and a fully-specified processing theory will necessarily specify their mode of use – where they originate, where they may terminate, how many may be in play at any given time. What is perhaps more controversial is that these are essentially linguistic claims that make contact with relativization facts like those in 4.

That is to say, an adequate sentence processing model for English will distinguish the processing of English and non-English sentences like 4c and 4b. If readers do not, in fact, consider “to the boy” a legitimate perceptual option for the gap position “...explained the answer to GAP” the reason must lie in the difference between 4b and 4a where it is an option. Whatever this reason is, some method of encoding the non-possibility of an overt noun phrase terminating a relativization dependency is needed, and this method will constitute a particular position on linguistic structure.

3.5.2 Kayne

The promotion analysis is a particular position on relativization dependencies. The analysis, which dates back to the 1960s, is revived in Kayne (1994). For reasons having to do with Kayne’s general theory of phrase structure, he rejects the standard analysis of
relative clauses as adjoined noun modifiers, and proposes that, in a sentence like 4c, the underlying form of the subject is akin to 5.

(5)  \( [\text{IP the father explained the answer to } \text{[DP[+wh] who boy[+f]]} ] \)

According to Kayne, at an early stage (5) of syntactic derivation, the determiner phrase (DP) “who boy” occupies what will eventually be the gap position. This DP moves to a specifier position of the enclosing, empty-headed (C0) complementizer phrase (CP), thereby checking a feature +wh as indicated in 6.

(6)  \( [\text{CP [DP who boy[+f]]}, \text{C0 [IP the father explained the answer to } t_i ] ] \)

In a second movement, “boy” evacuates from DP, moving to another specifier (perhaps that of the silent agreement morpheme, Agr) as in 7—checking a different feature, +f.

(7)  \( [\text{AgrP boy}_j \text{ Agr [CP [DP who t}_j \text{]}, \text{C0 [IP the father explained the answer to } t_i ] } ] ] \)

The entire structure becomes a complement of a determiner to yield a larger DP in 8.

(8)  \( [\text{DP the [AgrP boy}_j \text{ Agr [CP [DP who t}_j \text{]}, \text{C0 [IP the father explained the answer to } t_i ] ] } ] ] \)

No adjunction is used in this derivation, and, unconventionally, the leftmost “the” and “boy” do not share an exclusive common constituent. Nor is the wh-word “who” co-indexed with anything. Structural descriptions involving both the Kaynian analysis and the more standard adjunction analysis are shown in figures 3.9 and 3.10 respectively. The other linguistic assumptions suggested by these diagrams are discussed in chapter 4.

### 3.5.3 Minimalist Grammars of relativization

With the non-concatenative ‘move’ rule, MGs can express transformational accounts of linguistic structure quite naturally. The Kaynian relative clause analysis shown in 3.9 can be implemented using movement features in several related lexical entries. Consider the entries for common nouns in 9.

(9)  a. \( [\text{boy}] :: [\text{n}] \)

     b. \( [\text{boy}] :: [\text{n}, -f] \)

This pair of lexical entries expresses the idea that “boy” has an optional promotion-licensee feature –f. Movement to check this promotion feature can be triggered by the corresponding +f feature of “who” as shown in 10.

(10) \( [\text{who}] :: [\text{‘Num’, +f, d, -case, -wh.rel}] \)
The first feature of the lexical entry in 10 is a selection feature for a Num phrase, such as “boy” (singular) or “boy’s” (plural). Merging “who” and “boy” deletes this selection feature and results in a structure 11 where “who” is the head and the +f is the leftmost feature.
(11) 
<
+\text{f}, \text{d}, \text{-case}, \text{-wh}_{\text{rel}} \text{ who} \quad \text{-f boy}

The structure 11 meets all the conditions for the move rule to apply. The “boy” subtree is replaced by a single node without features, and a new specifier is created. The result is shown in 12.

(12) 
>
\text{boy} <
\text{d}, \text{-case}, \text{-wh}_{\text{rel}} \text{ who} \quad \emptyset

The movement depicted in 11 and 12 implements the rearrangement that Kayne suggests happens at Agr. In the formal grammar, this string-rearrangement actually occurs earlier in the derivation than WH-movement does (cf. 6). However, the essential content of the proposal – that the base word order is “who boy” – is preserved. Similarly, the move rule also applies to a \text{-wh}_{\text{rel}} subtree. This movement is triggered by a licensor feature \text{+wh}_{\text{rel}} on the complementizer. Lexical entry 13 specifies that the phonetic content of the silent complementizer is the null string.

(13) \llbracket \text{t, +wh}_{\text{rel}}, \text{c}_{\text{rel}} \rrbracket

Lexical entry 13 selects a tensed phrase, attracts a \text{-wh}_{\text{rel}} subtree to its specifier, and produces a relative CP. The WH-movement of the re-arranged DP “boy who” to specifier of CP occurs in exactly the same way the move rule transforms 11 into 12.

The adjunction analysis is less complicated: the grammar simply states that a relative CP can right-adjoin to DP using the adjunction axiom in 14.

(14) \text{[d]} \llbracket \text{c}_{\text{rel}} \rrbracket

Section 3.7 compares processing predictions from grammars that are identical except for their treatment of relative clauses. The method used to derive these predictions are explained in section 3.6.

3.6 Procedure

Seeking an explanation for S. Hawkins and Keenan’s finding of increased difficulty along the AH, a MG was constructed to cover their experimental stimuli. The grammar contains only 52 types of lexical entries that differ in any way other than their phonetic features. This grammar improves on the proposal of J. Hawkins by specifying, among other
things, how dependencies between fillers and gaps are enforced. As discussed in section 3.5, two variants of the grammar were created, one adopting a relative clause analysis due to Kayne (1994) and the other adopting the more standard adjunction analysis. Further details on the grammar can be found in chapter 4.

Derivation trees were obtained for the twenty-four sentences at all Subject, Object, Oblique and Genitive positions on the AH using a parser described in appendix A. Branches of these derivation trees were viewed as PCFG rules with probabilities set according to the usual relative-frequency estimation technique (Chi, 1999). However, because the stimuli were intentionally constructed to have exactly four examples of each structure, these sentences were weighted in accordance with a corpus study (Keenan and Hawkins, 1987) to make their frequencies more realistic.\textsuperscript{12}

<table>
<thead>
<tr>
<th>percent RCs</th>
<th>grammatical relation</th>
<th>bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>46.16</td>
<td>subject</td>
<td>1</td>
</tr>
<tr>
<td>23.73</td>
<td>direct object</td>
<td>2</td>
</tr>
<tr>
<td>14.92</td>
<td>oblique and indirect object</td>
<td>3</td>
</tr>
<tr>
<td>5.00</td>
<td>genitive</td>
<td>4</td>
</tr>
<tr>
<td>0.00</td>
<td>other</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 3.11: Results of ‘Variation in UG’ corpus study, $N = 2238$

Since the raw counts are at a coarser grain than the classes used in psycholinguistic experiment, the results in figure 3.11 were interpolated using a quadratic model\textsuperscript{13}, shown in 3.2.

$$68.15 - 25.57bin + 2.41bin^2.$$ (3.2)

The unobserved cases were assumed to fall into bins that did not happen to have been sampled.

The derivation-PCFG\textsuperscript{14} for the 24 test sentences is given in figure 3.13.

\textsuperscript{12}The Keenan and Hawkins (1987) corpus data were hand-counted. Accurate automatic labelling of the grammatical relation borne by an NP to a verb – even from parsed corpora – will probably require the resolution of many subproblems, including PP attachment ambiguities, recognition of the government relation, and detection of certain local transformations (Hindle and Rooth, 1993; Schmid, 2001; Gildea and Jurafsky, 2002).

\textsuperscript{13}The quadratic model fits the frequency data rather better $r^2 = 0.98$ than does a linear model $r^2 = 0.90$.

\textsuperscript{14}Formally speaking, this theory has 29 numerical parameters. Since $29 > 6$, it is certainly not a compact description of the AH performance data or the empirical frequency distribution of relative clause types in text. Despite being nonminimal for this data set, the grammar generalizes aggressively, making many other predictions not tested here regarding the existence and relative frequency of other sentence types. Extending it by adding more lexical entries is comparatively straightforward. Indeed, as discussed in 3.1.1, figure 3.13 is better thought of as the machine-readable version of an auxiliary hypothesis about the form of hearers’ mental grammars that happens to be an inefficient coding of six frequency facts.
<table>
<thead>
<tr>
<th>interpolated percentage</th>
<th>grammatical relation</th>
<th>bin</th>
</tr>
</thead>
<tbody>
<tr>
<td>45.0</td>
<td>subject</td>
<td>1.0</td>
</tr>
<tr>
<td>26.7</td>
<td>direct object</td>
<td>2.0</td>
</tr>
<tr>
<td>13.13</td>
<td>indirect object</td>
<td>3.0</td>
</tr>
<tr>
<td>8.18</td>
<td>oblique</td>
<td>3.5</td>
</tr>
<tr>
<td>4.43</td>
<td>genitive subject</td>
<td>4.0</td>
</tr>
<tr>
<td>1.90</td>
<td>genitive object</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Figure 3.12: Interpolated weights used to train PCFG

Since the grammar treats tense and other aspects of the English auxiliary system, members of the test set were assumed to be the result of a morphological analyzer (perhaps $M_1$) that has segmented the words of the actual stimuli. For example, “wrote” is segmented into “write -ed” “was” is segmented into “be -ed” et cetera.

For each sentence in the 24-member test set (see figure 4.1) the entropy reduction at each prefix was calculated. This was done by chart-parsing a regular expression consisting of the prefix string followed by any terminal, any number of times\(^{15}\). The entropy of this chart, viewed as a PCFG, was then calculated by evaluating equation 3.1. Downward changes in these values were then summed for each sentence to arrive at prediction of total difficulty.

3.7 Results

The summed entropy reductions, in general, confirm the Entropy Reduction Hypothesis (hypothesis 1) by correctly predicting increasing reading time farther out on the AH. Nonetheless there are four striking outliers in the raw results (not depicted):

the fact that the girl who play -ed for the ticket -s be -s very poor doesn't matter
he remember -ed that the sweet -s which David give -ed Sally be -ed a treat
they have -ed forget -en that the box which Pat bring -ed the apple -s in be -ed lost
he remember -ed that the girl who s mother send -ed the cloth -s come -ed too late

These sentences are just the ones that happen to have plural noun phrases. In essence, the stimuli were not balanced for plurality and this becomes a highly informative part of the string. Since the AH, and not plurality motivated this study, the test set was altered so that all noun phrases were singular. Training on this slightly altered set of sentences, the entropy reduction predictions do exhibit significant correlations with both the AH (figure 3.14) and

\(^{15}\)To make this process practical, only derivation tree branches observed in correct parses of the test set were ever considered. This artificially restricts the set of continuations to those that can be analyzed using the derivation-PCFG. Since no MG treebanks exist, statistics on other derivation tree branches are unavailable, and the effect of this approximation cannot be accurately determined.
Figure 3.13: Derivation-PCFG for Keenan and S. Hawkins test set
with subjects’ error scores (figure 3.15). A linear relationship 3.3 is proposed between the error scores and the entropy reductions.

\[
25.09 \text{ errors} + 0.05 \text{ errors per bit.} \quad (3.3)
\]

Simple log-probabilities of the derivation for stimulus sentences do not correlate as well with AH position (figure 3.16). Nor do similar entropy reduction predictions (figure 3.17) drawn from an alternative grammar similar to the one described in chapter 4 but employing the more standard adjunction analysis of relative clauses. These results show that the structure of the grammar is important for the predictions, and that this particular grammar is itself a rather poor memorizer of the frequencies on which it was trained.

However, the particular probability assignment is also important; there is no correlation between the prediction of a equiprobable-weighted grammar and the AH (figure 3.18). Indeed, confirming J. Hawkins’ original intuition, the average number of derivation tree nodes does match up well with AH position (figure 3.19).

### 3.8 Discussion

Hypothesis 1 generates sentence-level predictions that correlate significantly with both the ordering proposed by Keenan and Comrie (1977) and the error scores collected by Keenan and Hawkins (1987). The latter data were measurements of whole-sentence difficulty, and in this regard, hypothesis 1 explains the phenomena in terms of a generative
Figure 3.15: Entropy reduction vs. subjects' error scores

Figure 3.16: Log probabilities vs. position on AH
Figure 3.17: Entropy reductions on adjunction-analysis grammar vs. position on AH

Figure 3.18: Entropy reduction on a \textit{uniformly}-weighted grammar vs. position on AH
grammar and information theory.

However, hypothesis 1 is more specific than it needs to be in order to explain this sentence-level data. It also provides word-by-word predictions. At this resolution, the predictions are likely to be incorrect in detail. For instance Grodner, Watson, and Gibson (2000) found that, when other factors are well-controlled, the difference between subject- and object-extracted relative clauses resides primarily on the embedded verb. Comparing the predictions made by the grammar of chapter 4 on the subject- and object- extracted sentences of the AH test set, these predictions are only sometimes upheld. Depending on the sentence, spurious reading time peaks are also predicted. The bar graphs of figures 3.20-3.25, if only augmented with a linking theory between bits and reading time, would display all the predictions.

Rather than disconfirming hypothesis 1, the wide variability across particular sentences suggests that the Keenan-S. Hawkins test set (and the grammar that covers it) involves many uninteresting details from the point of view of the AH. The singular/plural issue discussed in section 3.7 exemplifies the problem. To study the AH in more detail requires stimuli that control these factors, which are often essential from the perspective of linguistic adequacy or formal completeness.

To that end, chapter 5 reports the result of three psycholinguistic studies of core AH properties: genitivity (5.1), indirect vs. directness (5.2) and obliqueness (5.3).
Subject-extraction

Figure 3.20: Word-by-word entropy reductions for subject extraction
Direct Object-extraction

Figure 3.21: Word-by-word entropy reductions for DO-extraction
Indirect Object-extraction

Figure 3.22: Word-by-word entropy reductions for IO-extraction
Oblique Object-extraction

Figure 3.23: Word-by-word entropy reductions for oblique-extraction
Genitive Subject-extraction

Figure 3.24: Word-by-word entropy reductions for genitive subject-extraction
Genitive Object-extraction

Figure 3.25: Word-by-word entropy reductions for genitive object-extraction
3.9 Conclusion

Viewed as a contribution to the research program that Miller and Chomsky (1963) and Stevens and Rumelhart (1975) pursued, the results of this chapter suggest a variety of conclusions.

Most narrowly, an explanation for increasing processing difficulty along the Keenan and Comrie (1977) Accessibility Hierarchy has been proposed. The explanation is that the syntactic structure of – for example – Genitive Object-relativized noun phrases – is more uncertain during incremental comprehension than the structure of, say, Subject-relativized noun phrases. The plausibility of this explanation is supported by a correlation between predictions derived from it and repetition error scores, themselves a measure of understandability.

On the one hand, this demonstration formalizes and confirms the intuitive proposal of J. Hawkins (1994). But on the other, it suggests new methods that are significant. The chapter details how it is technically possible to combine the idea that reading time is proportional to information processing effort with identifiable grammatical proposals from linguistics. In doing so, it shows how to follow the Competence Hypothesis far enough to arrive at quantitative predictions of linguistic performance. The prescription is to integrate existing techniques due to Grenander, Lang, Stabler and Harkema to calculate the change in the entropy of the start symbol of a probabilistic grammar as a given prefix string is lengthened. Compared to the method of chapter 2, these techniques not only provide a better connection to existing formal language theory but also generalize to a mildly-context sensitive grammar formalism. They are significant because psycholinguistic hypotheses respecting the Competence Hypothesis have greater scientific value than those that do not.

Indeed, if the processing hypothesis 1 continues to be supported, then processing evidence might be brought to bear on issues usually considered exclusively grammatical. The entropy reduction predictions from the relativization-as-adjunction grammar do not correlate with the Accessibility Hierarchy, while those from the grammar employing the Kaynian analysis do. Perhaps the Kaynian analysis and not the standard adjunction analysis ought to be adopted in a realistic transformational grammar. While that conclusion is of course somewhat tentative, it cannot be denied that a new method for systematically evaluating a particular processing hypothesis on a wide range syntactic proposals has been presented.
3.10 Appendix: Consistency Issues

Following Billot and Lang (1989), section 3.2.2 suggested a way to obtain a new PCFG that generates all and only the grammatical continuations of a given prefix. Rules in this new chart PCFG are defined by the presence of back pointers from chart-cell to chart-cell. Probabilities from the original grammar can be carried over from matching rules, e.g. the situated rule \( \text{NP}_{(2,5)} \rightarrow \text{D}_{(2,3)} \text{ N}_{(3,5)} \) would initially be assigned the probability of the unsituated rule \( \text{NP} \rightarrow \text{D N} \) from the original grammar. In the presence of ambiguity, there is no in-principle limit to the number of situated rules in the chart sharing the same left hand side, so the chart PCFG needs to be re-normalized to ensure that the distribution on alternative right hand sides is proper. This can be done straightforwardly, by dividing the probability of each rule whose lefthand side is \( \text{NP}_{(2,5)} \) by the total probability of all such rules.

However, even such re-normalization may not alone be sufficient to create a consistent probability model. The danger, identified in section 2.6, is that even a PCFG normalized in this way may assign probability to infinite derivation trees. Intuitively, this is because the decision to stop growing a derivation tree (by rewriting all nonterminals as terminal symbols) is defined locally and remains that same even as trees get bigger and bigger. If this stopping probability is too low, derivation can go on forever, sapping probability mass away from derivations that do eventually end. This results in a defective probability model of the language because the sum of the probability assigned to all sentences is less than 1.

To head off this danger, a technique known as Exponential Tilting (ET) can be employed.

3.10.1 Exponential Tilting

Exponential tilting is a well-known (Athreya and Ney, 1972, 48,52) technique in the branching processes community. PCFGs can be viewed as branching processes whose state is a vector of integers. Each component of the vector records how many symbols of each type (S,VP,D,CP...) remain in the sentential form being rewritten. Since rewriting occurs independently for each nonterminal, only the number (and not the tree-geometric location) matters. The initial state is a vector of the form \( \langle 0 \ldots 1 \ldots 0 \rangle \) corresponding to a single instance of the grammar's start symbol. At successive (discrete) time steps, symbols of each type are randomly rewritten according to the probabilistic phrase structure rules of the grammar. Nonterminals resulting from such rewriting are tabulated to determine the next state vector. It is called extinction when all nonterminals of some type happen to be rewritten as terminals. If all types are extinguished, then the process as a whole is said
to have gone extinct. If extinction is certain, the process is subcritical, if uncertain it is supercritical. PCFGs specify multitype Galton-Watson processes (named for the individuals who first used them to study extinction of family names in the British peerage) because different nonterminals can have different offspring distributions depending on the grammar. The word branching itself means that rewriting rules apply to each instance of a nonterminal symbol separately, potentially creating branches just as in a geneological family tree. Theoretical work on branching processes often deals with the singletype case in which the vector of types has only one component. The presentation in this appendix begins with the singletype case (i.e. a grammar with only one nonterminal that is also the start symbol) and then generalizes to the multitype case. More information on multitype branching processes can be found in Mode (1971).

The idea of exponential tilting is to alter a branching process’ offspring distribution to make larger trees less likely. In terms of a PCFG, this means changing rule weights without altering any rules so that no claims about linguistic structure are altered.

Let $r$ be the amount of tilting in the interval $[0, 1]$. Let $p(c)$ be the probability of having $c$ children – there is only one variety of child in the singletype case. Let $f$ be the probability generating function$^{16}$ of the branching process, i.e.

$$f(t) = \sum_c p(c) \cdot t^c$$

where $t$ is a dummy variable (this is also the $Z$-transform of the $p(c)$ in the variable $t^{-1}$). Then the new, $r$-tilted probabilities of having $c$ children are

$$p_r(c) = \frac{p(c) \cdot r^c}{f(r)}$$

In equation 3.4, the probability of having $c$ children is scaled back by a factor $r^c$, which gets exponentially smaller as the number of children gets larger. The scaling in the numerator of equation 3.4 has the same form as the individual terms of the generating function $f$ (with $r$ substituting for $t$), so the sum of these numerator terms is $f(r)$. Dividing each scaled probability by $f(r)$ ensures that the tilted $p_r$ define a proper probability distribution.

Now, Harris (1963) shows (page 7) that if extinction is not certain, the extinction probability is the unique nonnegative solution less than 1 of the equation

$$s = f(s),$$

$^{16}$Generating functions are a commonly used tool for dealing with sequences, such as the sequence of probabilities associated with an integer-valued random variable (Feller, 1957, chapter XI). For their application to PCFGs, see (Thomason, 1990).
This solution, \( q_i \), is the probability that the PCFG generation process will terminate after a finite number of rewritings. Tilting by \( r = q \), and using Harris’ observation that \( q = f(q) \), the new probabilities are

\[
p_q(c) = p(c) \cdot q^{-1}
\]

These new probabilities define a branching process where extinction is a certainty. The following proof, communicated to this author by James A. Fill and further explained by Nevin Kapur, establishes the result in the more general multitype case that is relevant to (chart) PCFGs. As discussed in section 3.10.2, the preconditions of positive regularity and nonsingularity do not restrict the practical applicability of the theorem.

Extending the previous notation, the probabilities \( p \) become vectors \( g \) with components \( g_i \), each a function of the vector \( \bar{c} \) of potential parents of each type.

**Theorem 1** Let \( X \) be a positively regular and nonsingular strictly supercritical multitype Galton-Watson tree. Let \( q_i \) be the probability of extinction from a single particle of type \( i \) so that for all \( i \), \( q_i < 1 \). Assume \( q_i > 0 \) for all \( i \). Let \( g_i \) be the offspring probability mass function (pmf) for type \( i \).

For all \( i \), define a new \( q \)-tilted pmf \( \bar{g}_i \) by

\[
\bar{g}_i(\bar{c}) = g_i(\bar{c})q_i^{-1}, \ldots, q_i^{-r}/q_i
\]

[Indeed, as \( \bar{c} \) varies, the numbers \( \bar{g}_i(\bar{c}) \) are nonnegative, summing to

\[
f_i(q_1, \ldots, q_s)/q_i = 1
\]

where \( f_i \) is the probability generating function for \( g_i \). This is a direct consequence of theorem 7.1 of Harris (1963) which generalizes 3.5 to the multitype case.]

Let \( \bar{X} \) be the corresponding (positively regular and nonsingular) multitype Galton-Watson tree. Then \( \bar{X} \) is (weakly) subcritical; moreover \( \mathcal{L}_i(\bar{X}) = \mathcal{L}_i(X|X \text{ is finite}) \).

**Proof:**

Writing \( P_i(\cdot) \) for the probability of a Galton-Watson tree rooted in type \( i \), we will show that, for all starting types \( i \) and finite trees \( T \),

\[
P_i(\bar{X} = T) = P_i(X = T| X \text{ is finite}) = P_i(X = T)/q_i \quad (3.6)
\]

Because equal probabilities imply equivalence in law, the first equality in 3.6 establishes the
theorem. The proof proceeds by showing the last equality, before showing that the first and last terms are equal by induction.

Consider just the last equality of 3.6. This is an instance of the set-theoretic identity $A = (A \cap B) \cup (A \cap B^c)$, namely:

$$P_i(X = T) = P_i(X = T \mid X \text{ is finite}) \cdot P_i(X \text{ is finite}) + P_i(X = T \mid X \text{ is not finite}) \cdot P_i(X \text{ is not finite})$$

The event $X = T$ either occurs with $X$ being finite or infinite. And $P_i(X \text{ is finite})$ is simply the extinction probability $q_i$. Since only finite trees are under consideration, the set of $T$ such that $X = T$ and $X$ is not finite is empty, hence the first term, $P_i(X = T \mid X \text{ is not finite})$, is zero. We have

$$P_i(X = T) = P_i(X = T \mid X \text{ is finite}) \cdot q_i + 0 \cdot P_i(X \text{ is not finite}) = P_i(X = T \mid X \text{ is finite}) \cdot q_i$$

$$P_i(X = T)/q_i = P_i(X = T \mid X \text{ is finite})$$

(3.7)

Having shown 3.7, it will suffice to show that $P_i(\bar{X} = T) = P_i(X = T)/q_i$ to establish all of 3.6 and hence the theorem. Such a demonstration also suffices for the subcriticality of $\bar{X}$, since by definition of conditional probability

$$1 = \sum_{T \text{ finite}} P_i(X = T \mid X \text{ is finite})$$

and using 3.7

$$1 = \sum_{T \text{ finite}} P_i(X = T)/q_i = \sum_{T \text{ finite}} P_i(X = T \mid X \text{ is finite})$$

$\bar{X}$ is subcritical if, for each finite $T$, $P_i(\bar{X} = T) = P_i(X = T)/q_i$. This is because, if all probability mass will be used up replicating the assignment of $P_i(X = T)/q_i$ to the finite trees, then there will be no mass left over to assign to any infinite trees. All that remains, therefore, is to show, $\forall i, T$ finite that $P_i(\bar{X} = T) = P_i(X = T)/q_i$. The proof is by induction on the size of $T$.

**BASE CASE**
When $T$ comprises only the root of type $i$ we have

$$P_i(X = T) = \hat{g}_k(\bar{0}) = g_k(\bar{0})/q_i = P_i(X = T)/q_i$$

**Inductive Step**

Then, denoting by $p_i$ ($\hat{p}_i$) the (tilted) probability that a node of type $i$ is individually extinguished, and identifying as $c_i$ the number of nodes of type $i$ on the frontier of a tree, we have

$$P_i(X = T) = \hat{g}_k(\bar{c})\hat{p}_1(T_{11}) \cdots \hat{p}_1(T_{1c_1})
\times \hat{p}_2(T_{21}) \cdots \hat{p}_2(T_{2c_2})
\cdots
\times \hat{p}_s(T_{s1}) \cdots \hat{p}_s(T_{sc_s})
= g_k(\bar{c})q_1^{c_1} \cdots q_i^{c_i-1}p_1(T_{11})/q_1 \cdots p_1(T_{1c_1})/q_1
\times p_2(T_{21})/q_2 \cdots p_2(T_{2c_2})/q_2
\cdots
\times p_s(T_{s1})/q_s \cdots p_s(T_{sc_s})/q_s
= q_i^{-1} g_i(\bar{c}) p_1(T_{11}) \cdots p_s(T_{sc_s})
= P_i(X = T)/q_i$$

as claimed.

**3.10.2 Application to chart PCFGs**

Since the probability of finite trees generated by a $q$-tilted PCFG is the same as probabilities assigned by the PCFG before tilting, the entropy of the finite derivations is also necessarily the same. Exponential tilting is hence a general method for bringing chart PCFGs to consistency for purposes of calculating the entropy of grammatical continuations.

Although the theorem is strictly only applicable to positively regular processes (the implication for PCFGs would be that all nonterminals are derivable from all other nonterminals, which is rarely desirable) the condition of positive regularity is only imposed so that $\bar{q}$ is well-defined as the fixpoint of the probability generating function $\bar{g}$. Even if the process is not positively regular, as in restrictive grammars, $\bar{q}$ can still be determined numerically to any degree of precision by repeatedly evaluating $\bar{g}$.

Singularity, which could only occur in a case where each chart PCFG rule is unary, exposes a menagerie of special cases in branching processes (Athreya and Ney, 1972, §9).
These can be avoided by tilting even farther than $r = q$.

3.11 Appendix: derivation of the DO/IO prediction

The set of parse trees consistent with a given prefix is often very larger, even in a restrictive grammar such as the one described in chapter 4.

This subsection illustrates how one implication of the AH — that direct-object extraction is easier to process that indirect-object extraction — can be derived by hand, by appealing to probability distributions on the rules of a realistic probabilistic grammar. The prediction itself is tested and found to be false in chapter 5, however the derivation itself is instructive in understanding the general theory.

For simplicity, assume a kind of GPSG (Gazdar et al., 1985) producing structural descriptions as in figures 3.26 and 3.27.

The entropy reduction hypothesis 1 can derive a prediction of greater difficulty
of the IO-extraction over the DO-extraction under various grammatical and statistical assumptions detailed below.

**Hypothesis 1 (Entropy Reduction Hypothesis)** A person's reading time at a word in a sentence is linearly related to any downward change in the entropy of the set of derivations generating the observed words as a prefix.

The fundamental idea is that, in contexts where an indirect-object relativization dependency is in play, the number of arguments an embedded verb has is more uncertain than it is elsewhere.

**Verb phrase rules**

The most radical simplifying assumption is that the embedded verbs in these contexts have just three possible subcategorization frames:

\[ _\text{NP} \quad _\text{PP} \quad _\text{NP} \quad _\text{PP} \]

Counterexamples to this assumption, such as “The oracle revealed [CP that the answer was 42 ]” are easy to find, however, some subcategorization restriction will need to be made, and this one facilitates the use of corpus data to be discussed presently.

Perhaps a more plausible assumption is that, on a realistic grammar, the prefix string “the secretary who the student...” (from condition B) is unambiguous. This situation is depicted in figure 3.26. Having heard only these words, the internal structure of the subtree rooted in VP/NP remains undetermined. The presence of the slashed category NP means that the subcategorization of the embedded verb must include an NP. What is unknown is whether the continuation of the sentence in 3.26 will be, for instance, “…sent is great” or “…sent to the principal is great.”

In terms of phrase structure this question is the decision between the rules

\[ \text{VP/NP} \rightarrow \text{V NP/NP} \]

and

\[ \text{VP/NP} \rightarrow \text{V NP/NP PP} \]

A lexically-specific property, subcategorization is better handled using *lexicalized* grammar rules (Schabes, 1990; Carroll and Rooth, 1998).

**Subcategorization preferences**

Estimating subcategorization probabilities is a notoriously messy problem. One of the sub-problems is that, especially in phrase-structure annotations schemes, subcategorized PPs like “to the principal” are often marked up in the same way as adjunct PPs like “on Tuesday.”
<table>
<thead>
<tr>
<th>verb</th>
<th>- NP</th>
<th>- PP</th>
<th>- NP PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>introduced</td>
<td>50</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>passed</td>
<td>32</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>presented</td>
<td>8</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>provided</td>
<td>48</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>returned</td>
<td>7</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>surrendered</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.1: Subcategorization counts adapted from Hockenmaier

0.16 VP/NP $\rightarrow$ introduced NP/NP
0.01 VP/NP $\rightarrow$ introduced NP/NP PP
0.16 VP/NP $\rightarrow$ passed NP/NP
0.01 VP/NP $\rightarrow$ passed NP/NP PP
0.11 VP/NP $\rightarrow$ presented NP/NP
0.06 VP/NP $\rightarrow$ presented NP/NP PP
0.13 VP/NP $\rightarrow$ provided NP/NP
0.04 VP/NP $\rightarrow$ provided NP/NP PP
0.13 VP/NP $\rightarrow$ returned NP/NP
0.04 VP/NP $\rightarrow$ returned NP/NP PP
0.13 VP/NP $\rightarrow$ surrendered NP/NP
0.03 VP/NP $\rightarrow$ surrendered NP/NP PP

Figure 3.28: grammar of VP/NP

The estimation method used here takes advantage of a version of the Penn Treebank that has been converted to Combinatory Categorial Grammar (CCG) (see Steedman (2000)). A lexicalist formalism like CCG encodes the subcategorization frame on the preterminal label of a verb. This makes it easy to count subcategorization frames. Counts of CCG lexical entries in the Wall Street Journal (WSJ) courtesy of Julia Hockenmaier\textsuperscript{17} were manually matched with the subcat frames under consideration.

Only a handful of verbs were both used in experiment 2 and occur in the WSJ in all the relevant subcategorization frames. This will be important for comparing the same verb across both conditions.

Adopting two more assumptions that are also surely false,

**flatness of head-choice distribution** the choice of head for VP is uniform

**context-freeness of subcat distribution** the choice of subcategory is the same in extracted contexts as everywhere else

then the probabilistic grammars of VP/NP and VP/PP in figures 3.28 and 3.29 encode the subcategorization information in table 3.1.

As is standard in GPSG, PP/PP and NP/NP derive only the empty string $\epsilon$.

\textsuperscript{17}This lexicon analyzes 48,844 sentences comprising 99.25\% of the Treebank
0.10 VP/PP $\to$ introduced PP/PP
0.07 VP/PP $\to$ introduced NP PP/PP
0.12 VP/PP $\to$ passed PP/PP
0.04 VP/PP $\to$ passed NP PP/PP
0.12 VP/PP $\to$ presented PP/PP
0.05 VP/PP $\to$ presented NP PP/PP
0.07 VP/PP $\to$ provided PP/PP
0.09 VP/PP $\to$ provided NP PP/PP
0.14 VP/PP $\to$ returned PP/PP
0.03 VP/PP $\to$ returned NP PP/PP
0.08 VP/PP $\to$ surrendered PP/PP
0.08 VP/PP $\to$ surrendered NP PP/PP

Figure 3.29: grammar of VP/PP

Deriving the predictions

To evaluate the Entropy Reduction hypothesis (1) it is necessary to calculate the change in the entropy of the set of derivations generating the observed words as a prefix.

By context-freeness and the earlier assumption of prefix unambiguity, the VP daughter of the root S (the main verb phrase) is not constrained in any way by the revelation of the embedded verb. So, only the derivation of VP/NP in the DO case and VP/PP in the IO case needs to be considered. Looking at just the DO case, this is the difference between $H(\text{VP/NP})$ and a term $H_5$, the entropy of the parser state having heard the 5th word.

Let the set of production rules in a probabilistic context-free grammar be $\Pi$ and the subset rewriting nonterminal $\xi$ be $\Pi(\xi)$. Denote by $p_r$ the probability of a rule $r$. Then

$$h_i = h(\xi_i) = - \sum_{r \in \Pi(\xi_i)} p_r \log_2 p_r$$

$$H(\xi_i) = h(\xi_i) + \sum_{r \in \Pi(\xi_i)} p_r [H(\xi_{j_1}) + H(\xi_{j_2}) + \cdots] \quad (3.8)$$

(Grenander, 1967, 19)

Equation 3.8 gives the entropy for a nonterminal in a probabilistic context-free grammar. Consider first $H(\text{VP/NP})$. This is the sum of the single-rule entropy $h$ and the expected entropy of any nonterminal symbols appearing on the right hand side. But since slashed categories of the form $X/X$ never have any internal structure their entropy will always be zero. The only nonterminal that could matter in this case is PP. Let $\alpha$ range over right-hand sides (RHSes) of rules in grammars 3.28 and 3.29.
DO extraction:
introduced \(2.7160 + 0.1686 \times H(PP)\) bits
passed \(2.7356 + 0.1730 \times H(PP)\) bits
presented \(2.6152 + 0.1225 \times H(PP)\) bits
provided \(2.6398 + 0.1404 \times H(PP)\) bits
returned \(2.6409 + 0.1410 \times H(PP)\) bits
surrendered \(2.6479 + 0.1447 \times H(PP)\) bits

Table 3.2: Direct Object extraction

IO extraction:
introduced \(2.8777 + 0.2974 \times H(NP)\) bits
passed \(2.9067 + 0.3272 \times H(NP)\) bits
presented \(2.8935 + 0.3175 \times H(NP)\) bits
provided \(2.8770 + 0.2755 \times H(NP)\) bits
returned \(2.9336 + 0.3410 \times H(NP)\) bits
surrendered \(2.8752 + 0.2855 \times H(NP)\) bits

Table 3.3: Indirect Object extraction

\[
H(VP/NP) = h(VP/NP) + \sum_{\alpha \text{ contains PP}} p_\alpha \cdot H(PP)
\]

When the embedded verb is revealed, the structural possibilities of \(VP/NP\) become more constrained. The compatible rules must match their first RHS symbol \(\alpha_0\) with the embedded verb \(w_5\).

\[
H_5 = \sum_{\alpha_0 = w_5} p_{VP/NP \rightarrow \alpha} \log_2 p_{VP/NP \rightarrow \alpha} + p_{VP/NP \rightarrow w_5, NP/NP \rightarrow PP} \cdot H(PP)
\]

Manipulating subscripts, expressions for the entropy differences can be given.

\[
H(VP/NP) - H_5 = \sum_{\alpha_0 \neq w_5} p_{VP/NP \rightarrow \alpha} \log_2 p_{VP/NP \rightarrow \alpha} + \sum_{\alpha_0 \neq w_5, PP \in \alpha} p_{VP/NP \rightarrow \alpha} \cdot H(PP)
\]

\[
H(VP/PP) - H_5 = \sum_{\alpha_0 \neq w_5} p_{VP/PP \rightarrow \alpha} \log_2 p_{VP/PP \rightarrow \alpha} + \sum_{\alpha_0 \neq w_5, NP \in \alpha} p_{VP/PP \rightarrow \alpha} \cdot H(NP)
\]

Using the values from the grammars in figures 3.28 and 3.29 the entropy reductions are indicated in tables 3.2 and 3.3 respectively.

Summing over each of these idiosyncratic verbs, the embedded verb provides \(1.36 + (1.84H(NP) - 0.89H(PP))\) more bits after the IO prefix than after the DO prefix. For English, it is rea-
sonable to suppose that $H(PP)$ is only slightly greater than $H(NP)$ since the additional structure present in the former describes only the closed class of prepositions. In combination with hypothesis 1, this predicts greater reading time on the embedded verb in IO-extracted relatives than in DO-extracted relatives.
Chapter 4

The ‘Larsonian’ grammar

This chapter documents the ‘Larsonian’ Minimalist Grammar used in chapter 3. It surveys the theoretical background behind the specific analyses used in this grammar and summarizes key arguments from the literature. The Minimalist Grammar formalism itself is defined explicitly in a series of publications by various authors (Stabler, 1997; Stabler and Keenan, 2000; Michaelis, 1998; Harkema, 2001).

This grammar was written with the primary goal of covering 24 sentences used in a sentence-memory experiment conducted in 1974 (Keenan and Hawkins, 1987). These sentences are listed in figure 4.1.

Even casual examination of these sentences reveals that they exercise a fair bit of English syntax. Any grammar that covers these stimuli will have to reckon with perennial matters of linguistic debate, including the proper description of ditransitive verbs (section 4.1), relative clauses (section 4.3), case (section 4.4), tense and other details of the English auxiliary system (section 4.5), noun phrase issues such as agreement and the Saxon genitive ’s (section 4.2), and the distinction between arguments and adjuncts (section 4.6). Some justification for the analysis that was actually adopted will be given in the corresponding section, but the reader should bear in mind that the ultimate reason will always turn out to be “because it was simpler that way.” Simplicity guided the creation of the grammar and motivates the hope that any results derived from it will generalize to new sentences.

Familiarity with some notational conventions will aid in reading the actual grammar (appendix 4.8).
<table>
<thead>
<tr>
<th>symbol</th>
<th>meaning</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals sign</td>
<td>select feature</td>
<td>(=t) select a tense phrase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(=\text{little} _v) select light verb phrase</td>
</tr>
<tr>
<td>bare letter</td>
<td>category feature</td>
<td>(d) determiner</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(p _t_o) to-headed preposition</td>
</tr>
<tr>
<td>plus sign</td>
<td>attract feature</td>
<td>(+\text{case}) attracts (-\text{case}) phrase to cancel both under spec-head agreement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(+\text{f}) ‘focus’ feature promotes the head of a relative clause</td>
</tr>
<tr>
<td>minus sign</td>
<td>licensee feature</td>
<td>(-\text{wh}_\text{rel}) feature that motivates \text{WH}-movement</td>
</tr>
<tr>
<td>rightward</td>
<td>left incorporation</td>
<td>(=&gt;\text{little} _v) select a light verb phrase, but incorporate its head on the left side of the selecting item</td>
</tr>
<tr>
<td>arrow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>long right</td>
<td>right affix-hopping</td>
<td>(n=&gt;) genitive lowers onto its nominal complement</td>
</tr>
<tr>
<td>arrow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>left-opening</td>
<td>left adjunction</td>
<td>([\text{deg}]&gt;&gt;[\text{'}A\text{'}]) a degree modifier can adjoin to an adjective (phrase) on the left</td>
</tr>
<tr>
<td>French quotes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>right-opening</td>
<td>right adjunction</td>
<td>([v]&lt;&lt;[\text{tmp}]) a temporal modifier can adjoin to a verb (phrase) on the right</td>
</tr>
<tr>
<td>French quotes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Certain extensions to the basic Minimalist operations ‘Merge’ and ‘Move’ are applied in the grammar. These extensions provide the flexibility to express many analyses that have become standard in generative grammar (Stabler, 2002).
4.1 Ditransitive verbs

The grammar is named in honor of Richard Larson, whose influential 1988 article “On the Double Object Construction” provides the basis for the treatment of ditransitive verbs. The essential idea is that a ditransitive verb like sent has a tighter connection to an indirect object (IO) like to Mary than to a direct object (DO) such as a letter. Larson states it terms of constituency:

a simple dative like John sent a letter to Mary derives from an underlying form in which the verb and its indirect object make up a constituent that excludes the direct object.

(Larson, 1988, 335)

Other facets of derivation install the direct object and sentential subject John in the proper places. Larson points out that this claim conflicts (on the assumption of a certain binding theory) with previous proposals advocating flat structure, or a constituent pairing the verb and direct object but not the verb and the indirect object.

4.1.1 The rise and fall of Larson’s idea

In many ways, the grammar is mis-named because this idea is not really Larson’s at all. He attributes it to Chomsky (1955/1975) who considers it briefly in Chapter X of the Logical Structure of Linguistic Theory (LSLT). Discussing the verb class $V_f$ (which includes ditransitives such as give and asked) in section 104.2, Chomsky proposes the phrase structure rule

$$V_T \rightarrow V_f PP_a$$

This phrase structure rule licenses an “underlying constituent” that, for instance, send and to Mary would share. A transformation, $T_a$ maps between the underlying form and the derived form John sent Mary a letter\(^1\). In Syntactic Structures (1957), this analysis re-appears in discussion of particle movement. There (page 77) $T_a$ is renamed $T_{ap}^{\text{sb}}$ and, instead of PP, Chomsky deals with a nonfinite verbal complement studying in the library.

\(^1\)The underlying status of John sent a letter to Mary is supported by the appearance of the preposition to in questions assumed to be transformationally related:

who did John send a letter to?

to whom did John send a letter?
The analysis is nonetheless identical: in an item like (107), which is the underlying form of (106),

(107) John – found studying in the library – the boy
(106) John found the boy studying in the library

(Chomsky, 1957, 82)

Chomsky identifies *found studying in the library* as a “complex Verb.” This is the transitive verb $V_T$ lacking its direct object, having already combined with its indirect object.

Chomsky’s analysis is explicitly acknowledged in work within a Categorial Grammar framework (Bach, 1979, 519). In such a framework, an obligatory transformation (re-ordering direct and indirect objects) can be avoided by adopting a syntactic rule of “Right Wrap” as Dowty (1982) does.

... a three-place verb can be given the same relationship to a transitive verb as a transitive verb has to an intransitive verb...thus syntactically, a three-place verb will be combined with a term phrase to give a transitive verb.

In English, this rule [*the Verb-Indirect Object Rule – JTH*] inserts the preposition *to* before the term phrase, in other languages, it marks it with Dative case. I will call the category of ditransitive verbs “TV/T”. This means that the English Verb-Direct Object operation will have to be modified to a “Right Wrap” operation – that is, when combining a verb with a direct object, the direct object will be placed after the first word of the transitive verb phrase; the English sentences with such verbs will be produced as in the example (14). Fortunately, there is independent motivation for this Right Wrap operation in English, as has been shown by Partee (1975), myself (1979b: ch4), and Bach (1980).

(14) 

```
[John gives a book to Mary]_t
  |
John_T [give a book to Mary]_TV
  |
    [give to Mary]_TV
      |
        [a book]_t
          |
        give_TV/T
          |
            Mary_T
```

(Dowty, 1982, 89)

The topmost right branch of Dowty’s example (14), depicting the combination of *give to Mary* and a *book* is the branch where the Right Wrap operation applies. Its left daughter is the “underlying constituent” in the sense of Larson, where verb and indirect object are sisters.
In fact, Jacobson (1987) shows that Right Wrap can be reconstructed as slash-passing in GPSG (Gazdar et al., 1985)\(^2\) in a “Verb promotion” analysis. She proposes a “double slash” feature whose passage respects bounding nodes. Under this proposal, the indirect object would share a constituent with a phonetically null element connected by (double) slash-passing to the surface verb. Part of the beauty of the categorial analysis, implemented either by a nonconcatenative operation like Right Wrap or by Verb Promotion via slash features, is that successive arguments come into the derivation in reverse order of obliqueness: indirect object, then direct object, then subject. This insight is preserved in the HPSG Hierarchical Theory of Subcategorization (Pollard and Sag, 1987, 117). In this non-derivational theory, the ordering of complements by obliqueness visible in the categorial derivation is explicitly encoded by the order of items on SUBCAT lists in verbal lexical entries. However, a flat VP is adopted and constituency of the verb and indirect object is not asserted. This option is unavailable in a framework where every instance of syntactic composition necessarily implies the creation of new constituency relations – a property of many incarnations of the Minimalist Program.

4.1.2 Controversy over Minimalist approaches to ditransitives

Although Larson’s idea of an underlying constituent containing the verb and indirect object has been widely adopted (cf. references in (Baker, 1997)), there exists an opposing tradition which gained currency among Minimalist syntacticians. Confusingly identified (e.g. by Ura (2000)) as the “neo-Larsonian shell” approach, this view actually denies the most fundamental aspect of Larson’s analysis in supposing that it is the direct object that is in an underlying constituent with the verb. The essential phrase structure would be

\[
\begin{array}{c}
\text{VP} \\
\text{IO} \\
\text{V} \\
\text{DO} \\
\end{array}
\]

based on the c-command relationships implicated by a certain binding theory. One set of binding facts cited are from Icelandic (Collins and Thráinsson, 1996, 398) although many other languages including English are putatively covered by this analysis (Bobaljik, 1995).

This “neo-Larsonian” view is explicitly abandoned by Hale and Keyser (2002), who cite facts about the interpretation of secondary depictive predicates in English that

\(^2\)Jacobson also points out that Bach’s specific formulation of Right Wrap is empirically inadequate because it does not correctly handle conjoined arguments. The simplicity of this observation belies the profundity of the question it brings up, namely the amount and type of structural information that should be visible to rules of syntactic composition.
contradict earlier conclusions based on Icelandic binding facts. Hale and Keyser write,

While 15 embodies the hierarchical arrangement we have assumed for the internal arguments of the double-object construction (cf. Hale and Keyser 1993), and in fact that [sic] assumed for both the double-object construction and the to-dative alike in some recent proposals (e.g. Takano 1996) it is theoretically possible, of course that the hierarchical arrangement of arguments in 15 is wrong, for the double-argument construction at least.

*discussion of secondary deffective predicate facts elided – JTH*

If this is the general pattern, secondary predication is regularly of the theme, not the recipient, in both the to-dative and the double object construction (cf. Jackendoff, 1990:203; Rothstein, 1983).

... If the same is true in the double-object construction...then 15 is not (emphasis mine) the correct configuration for that construction. Instead a structure more closely akin to that proposed by Larson (1988) must be assumed – to wit, a configuration in which the theme (DO) is higher than the goal (IO), just as in the to-dative.

Baker had already remarked on the same disagreement several years earlier.

*[Baker's own] analysis crucially assumes that goals (IO) project into a lower structural position than themes (DO), in accordance with the Larson/Baker thematic hierarchy...If the goal were not generated lower than the theme, the theme could not prevent the goal from becoming a subject. Once again, my proposal is nearly the opposite of Everaert's (1990)...While the two approaches appeal to very much the same range of concepts, I believe that mine is correct because it generalizes better to other language families.

(Baker, 1996, 23)

### 4.1.3 A categorial-minimalist approach

My own approach follows Larson in a way considerably indebted to the categorial grammar tradition. A transitive verbal lexical entry

\[
[kiss] :: [=d, +case,v].
\]

takes one argument determiner phrase (DP) and 'assigns' it case^3 A ditransitive lexical entry is identical except for an additional argument.

\[
[give] :: [=p,to, =d, +case,v].
\]

The entry for *give* indicates that it selects a *to*-headed prepositional phrase (PP), then a DP, which it assigns case. As per the definition of 'merge' in Minimalist Grammars, the

---

^3Case-assignment under specifier-head agreement is one of the ways the grammar departs from the categorial tradition. See section 4.4.
first complement selected (i.e. IO) comes on the right, with successive merger creating new specifiers for DO and Subject on the left. This can be seen in figure 4.2 where the to-headed PP is the complement of the verb *give* and the *letter* occupies both specifiers of V. Movement to the second specifier is motivated by case-checking.

The structural description in figure 4.2 also exemplifies the mechanics of subject selection: the VP is selected by a light verb “little v,” becoming its complement. The light verb then merges the subject *John* as a specifier. Ultimately, the lexical verb *give* incorporates (a case of head movement) twice, from its base position between the DO and IO to little v to tense, the suffix *-ed*. Finally, the entire sentence is construed as being headed by an empty complementizer.

In summary, the analysis adopted for ditransitive verbs applies several ideas either popularized or originated by Larson, including the idea of VP-shells (like the one headed by little v in figure 4.2) and the idea that the V and the IO share a constituent. Straightforwardly taking advantage of the order of features in an MG lexical entry, the analysis also preserves the advantages of categorial, phrase structural and transformational accounts where arguments are composed with verbs in order of obliqueness.

### 4.2 Nominals: genitive and number agreement

The Larsonian grammar does not actually enforce number agreement. But the extension would be straightforward. In this grammar, common nouns (category n) are either unmarked, and hence singular, or marked with a suffix *-s* to indicate plurality.

\[
\begin{align*}
\text{ singular}: & \{=n, 'Num'?\} \\
\text{ plural} : & \{n=>>, 'Num'?\}
\end{align*}
\]

This is technically implemented by common nouns being complements of a category *Num*. The creation of such a number phrase is either via merge (=n) or right affix-hopping (=>>) which lowers the *-s* suffix onto the noun being number-marked. The category *Num* could be specialized to *Num*$_{plur}$ and *Num*$_{sing}$ or even *Num*$_{3rd-sing-fem}$, via lexical rules in the style of LFG (Bresnan, 1982b) – although this was not explicitly pursued. A serious treatment of morphology might be implemented as a finite state transducer that filters the sequence of lexical entries considered by a parser (Beesley and Karttunen, 2003).

Treatment of the genitive is similarly naïve. Following Abney (1987) (page 52) the Larsonian grammar takes the Saxon genitive *'s* to be a combiner of two (number-marked) noun phrases, one as complement and the other as specifier. Parting company with Abney, the category of the result is again that of a common noun, rather than a DP, i.e.
Such a structure builds recursion into the category n rather than the category d. It consequently suggests an interpretation in which determiner properties are predicated of the compound noun, e.g. the first rather than the second bracketing below:

\[
\begin{array}{c}
\text{girl's bike} \\
\text{unique of girls and their bikes} \\
\text{unique of girls only}
\end{array}
\]

Absent a more articulated account of determiner semantics, the adequacy of this syntactic analysis cannot really be evaluated. Its main virtue is compatibility with the idea of a "relative determiner" implemented in lexical entry 17 and required for the relative clause analysis discussed in section 4.3.

(16) \[ \text{the [girl's bike]} \]
(17) \[ \text{[the girl's bike]} \]

The relative determiner takes a relative clause as complement \( c_{\text{rel}} \) rather than a number phrase. Across all determiners, since genitive recursion has been isolated in n, the (single) d heading a DP can be a \(-\text{case} \) licensee, thereby setting up the requirement that (roughly) each full nominal must 'receive' case exactly once.

### 4.3 Relative clauses

The Larsonian grammar implements the raising or promotion analysis of relative clauses. Although this analysis is very old – cited as unpublished work of Michael Brane (1967) but independently promulgated in a host of less obscure places (Smith, 1964; Schachter, 1973; Vergnaud, 1974; Áfarli, 1994; Bhatt, 1999) – it has recently been revived by Kayne (1994) and Bianchi (1999) on whom this section leans heavily. The promotion analysis was adopted in order to explore the tradeoffs between Kayne's restrictive assumptions and other grammatical possibilities⁴.

The accepted view of relative clauses within the Chomskyan school is that they are noun modifiers. As such, the relation of modification that they bear to the relevant noun is indicated by Chomsky-adjunction to the right, depicted in 18 adapted from (Bianchi, 2000b, 53)

---

⁴A complementary grammar (the “Chomskyan” grammar) that analyzes relative clauses using adjunction has also been written.
In 18, the node CP is Chomsky-adjoined to the node NP. Since Chomsky-adjunction can occur on any category, the distinction between restrictive and non-restrictive

(19) restrictive: Mary knows few boys who enjoy knitting. $\nmid\neq$ Mary knows few boys.

(20) non-restrictive: Mary knows few boys, who enjoy knitting. $\models$ Mary knows few boys.

relative clauses can be conveniently notated by the bar level of the adjunction. Vergnaud (1974) suggested, for instance that non-restrictive relatives be adjoined at the double-bar (NP) level, with restrictives confined to the single-bar (N) level.

Kayne, however, rejects the adjunction analysis because right-adjunction is generally banned under his Linear Correspondence Axiom (LCA). The LCA is intended to “express the intuition that asymmetric c-command is closely matched to the linear order of terminals” (Kayne, 1994, 5). A version of Kayne’s technical statement of the idea is given below.

**Linear Correspondence Axiom** In a given phrase structure tree, let $A$ be the set of all pairs $(X, Y)$ of non-terminal nodes such that $X$ c-commands $Y$ but $Y$ does not c-command $X$. In the same phrase structure tree, let $d(X)$ be the set of terminal nodes dominated by a nonterminal $X$. Then the relation $(d(X), d(Y))$ for all $(X, Y) \in A$ is transitive, total, and antisymmetric.

In a structure such as 18 CP c-commands NP, “asymmetrically” because the top “segment” of NP dominates CP, preventing NP from symmetrically c-commanding CP back. So (CP, NP) is a member of the set $A$. Under the LCA, the yield of CP $\{\text{which, I, bought}\}$ is supposed to precede, not follow nor overlap with the yield of NP $\{\text{book, which, I, bought}\}$. But in 18 there is overlap between these two sets of terminals, and it is those terminals that constitute the antisymmetry violation\textsuperscript{5}. For instance $\text{book} \in d(CP)$ and $\text{which} \in d(NP)$, \textsuperscript{5}A relation $R$ is antisymmetric if the only symmetric relationship elements can be in is with themselves, i.e. for all $x, y, xRy$ and $yRx$ implies that $x = y$. For more information on relations see Partee, ter Meulen, and Wall (1993).
while symmetrically \textit{book} \in d(NP) and \textit{which} \in d(CP). So the adjunction analysis cannot be maintained in conjunction with the LCA.

In adopting the alternative promotion analysis, Kayne also re-interprets the wh-word not as a phrasal CP specifier but rather as a determiner of the “head” of a relative clause. At the beginning of the syntactic derivation, the wh-word and this head are adjacent. For example, a relative clause like 21

\begin{equation}
(21) \text{ the boy who I met }
\end{equation}

(Bianchi, 1999, 79)

has the derivation\textsuperscript{6} given in 22.

\begin{equation}
(22) \begin{aligned}
[\text{DP} \text{ the } [AgrD [CP I met [DP who [NP boy]]]]] \\
[\text{DP} \text{ the } [AgrD [CP [DP who [NP boy]] i [IP I met t_i]]]] \\
[\text{DP} \text{ the } [AgrP boy [AgrD [CP [DP who t_{NP}]] i [IP I met t_i]]]] \\
[\text{DP} AgrD + \text{the } [AgrP boy [t_{Agr} [CP [DP who t_{NP}]] i [IP I met t_i]]]]
\end{aligned}
\end{equation}

(Bianchi, 1999, 79)

The main difference between the Kaynian analysis and the more standard adjunction analysis is the fact that \textit{the} and \textit{boy} do not form a constituent. The underlying form is \textit{the} [\textit{CP I met who boy}] where [\textit{DP who boy}] moves to specifier of CP to check a +rel feature. Then \textit{boy} moves to the specifier of a silent AgrD head, which itself is incorporated with the external determiner \textit{the} upon their merger. This last instance of head movement is motivated by the need to check “strong nominal features” on AgrD.

Because nonrestrictives are distributionally so similar to restrictives, and the difference between them is essentially semantic, Kayne locates their difference at LF:

The parallel proposal that I would like to make for relatives is that restrictives and nonrestrictives differ at LF but do not differ structurally in the overt syntax.

(Kayne, 1994, 111)

\subsection{4.3.1 Evidence for and against the promotion analysis}

The original argument for the promotion analysis of relative clauses is from idiomatic expressions. Brame (1976) notes that a word like \textit{headway} is restricted to the fixed phrase \textit{make headway} as in 23.

\begin{footnote}{The Agr-based agreement system was popular in early Minimalist analyses but has since been abandoned (Chomsky, 1995b, §10).}
\end{footnote}
(23) We made headway.

The same property holds of other idioms\(^7\) such as keep careful track of, pay lip service to, take umbrage at et cetera. Crucially, the absence of the special verb make brings on unacceptability, as in 24.

(24) * (The) headway was satisfactory.

Item 24 is unacceptable presumably because it fails some subcategorization or other co-occurrence restriction. However, in a relative clause, headway can occur disconnected from make as in 25.

(25) The headway that we made was satisfactory.

Brame (1967) (quoted in Schachter (1973)) argues that 25 is grammatical because, at some stage of the derivation where the relevant subcategorization co-occurrence restriction was checked, headway indeed was a complement of make. The analysis is that headway has been transformationally promoted from a position adjacent to make, where it was base-generated, to its surface position between the determiner and the complementizer. Indeed, this is the same pattern of argument from co-occurrence restrictions that was used to establish a transformational relationship between active and passive sentences.

Kayne and Bianchi’s revival of the promotion analysis spawned a debate in the pages of Linguistic Inquiry. Some points of this debate are unique to the class of wh-relatives or the class of non-wh-relatives, and the data at issue are largely drawn from English. In many cases, the reply is simply to assume more silent functional categories and a longer syntactic derivation. Still, a subset of the contested points are briefly reviewed here to illustrate the character of the disagreement(s).

**Category of the trace**

Borsley (1997) charges that, if, in an example like 26

(26) \[ \text{DP the [CP [picture] [that [IP Bill liked t_i]]]]} \]

the relative head picture is promoted from the trace position t\(_i\) then items like 27

(27) * Bill liked picture

ought to be acceptable. But they are not. He goes on to show, using a variety of tests including binding, licensing of parasitic gaps and weak islandhood, that the trace t\(_i\) is actually a DP-trace, rather than an NP-trace. He concludes that Kayne’s analysis must be

\(^7\)See Vergnaud (1974) for parallel French examples.
revised such that the promoted element is a null-headed DP, rather than an NP. Bianchi (2000a) accepts this objection and adopts the structure [DP e [NP picture ] ] Since Bianchi is relatively unconcerned with the issue of proper government that lies behind Borsley’s criticism, this apparent concession actually sidesteps the issue.

**Stacking**

Borsley make essentially the same point in criticizing the extension of the promotion analysis to recursively embedded relative clauses like 28.

(28)  the [[ book ] that John wrote t_i ] that Bill burnt t_j

Item 28 depicts the phrase structure that Borsley assumes. His critique is that, if book that John wrote is a CP, headed by the complementizer that, then t_j ought to be a CP-trace. But the same evidence just cited in connection with 26 suffices to demonstrate that it is actually a DP-trace. So the same game must be played again, “Notice we now have two empty Ds and we need some mechanism to ensure that they are both empty” (Borsley, 1997, 637). Bianchi replies that, indeed, there are two phonetically-null determiners, and that an economy principle

(29)  Economy of Representation

Incorporate a functional head to a host whose feature structure is consistent with its own

triggers a kind of repair, namely deletion: “I propose that this is an effect of a double abstract incorporation: the lower D0 of DP_i incorporates to the higher D0 of DP_j, which selects the inner CP, and the resulting complex head incorporates to the highest external D0 the; both steps are triggered by the economy principle 29” (Bianchi, 2000a, 131). Here again the implicit accusation is that Bianchi is polluting the grammar with such gratuitous use of empty categories, and her answer is the grammar incorporates 29 and is responsible for cleaning itself.

**Coordination**

Borsley points out that under Kayne’s version of the promotion analysis, 30 violates any type of Coordinate Structure Constraint (e.g. (Ross, 1967)) since the conjoined strings are not, in fact, constituents.

(30)  the picture [ which Bill liked ] and [ which Mary hated ]

On Kayne’s analysis, the wh-determiner which heads a DP whose specifier includes picture and whose complement contains Bill liked. No individual constituents exist that encompass each of the conjoined elements themselves.
In response, Bianchi invokes an enriched analysis based on a “split-CP” hypothesis that she attributes to Rizzi (1997). In this enriched analysis, there is another silent functional head “in the Comp system below C0” (Bianchi, 2000a, 130). The wh-determiner resides in the specifier of this functional head, providing a constituent for the substring which Bill liked.

4.3.2 A simplified promotion analysis (without Agr)

The debate between Borsley and Bianchi suggests that a number of empirical issues may stand between the promotion idea and a fully general account of English relative clauses consistent with the LCA. Nevertheless, the Larsonian grammar adopts a simplified version of the Brame/Kayne/Bianchi analysis. In this version, common nouns come in two varieties – with and without an additional promotion feature. This is to say, expressions of category n are optionally promotion licensees endowed with the ‘prominence’ feature −f.

\[
\text{[boy]} :: [n]. \quad \text{[boy]} :: [n, -f].
\]

A Kaynian wh-determiner can take a number-marked promotion licensee as complement.

\[
\text{[who]} :: [=\text{‘Num’}, +f, d, -\text{case}, -\text{wh_rel}].
\]

Faithful to the promotion analysis, selecting a Num phrase yields, for example, who boy in situ.

\[
\begin{array}{c}
\text{DP} \\
\text{the} \\
\text{CP} \\
C^0_{+\text{wh_rel}} \\
\text{TP} \\
\text{the girl} \\
-\text{ed surprise} \\
\text{DP}_{-\text{wh_rel}} \\
\text{who}_{-f} \text{ boy}_{-f}
\end{array}
\]

Movement to specifier checks f and reverses the linear order of who and boy. After case-checking (see section 4.4), the entire DP boy who discharges its −wh_rel feature in movement to the specifier of a silent complementizer c_rel.

\[
\text{[]} :: [=t, +\text{wh_rel}, \text{c_rel}].
\]

In figure 4.3 the trace of the movement that checked f is labeled t(0); the trace of the movement that checked wh_rel is labeled t(1). This structure describes the object-extracted relative clause [DP the boy who the girl surprised].
Such a simplified version of the promotion analysis interacts harmoniously with the analysis of the genitive (section 4.2). As noticed by McDaniel, McKee, and Bernstein (1998) Kayne’s approach leaves the genitive marker ‘s string-adjacent to the wh-word who, clearing the way for a derivation \( \text{who} + \text{'s} \rightarrow \text{whose} \) in the morphology. This adjacency is shown in figure 4.4. This structure describes the subject-extracted relative clause [DP the girl whose cat liked the boy ].

4.4 Case and dative shift

The Larsonian grammar implements case-checking as movement to specifier. Under this proposal, DPs have a –case licensee feature, and the case-assigning categories verb, preposition and tense have +case attract features (Haegeman, 1991, 193). Stabler (2002) shows how such an arrangement can handle case-mismatch items involving raising and control verbs, as well as simpler constructions. In applying the idea of “null case” (cf. Chomsky and Lasnik (1995)) Stabler employs a second type of case feature. Such case-feature multiplication ought to extend straightforwardly to richer case-marking systems, perhaps distinguishing nominative, accusative or other cases. It is interesting to compare this derivational approach to case with the representational approach of Chomsky (1980).

(68) Suppose that Case Assignment follows these general principles:

- a. NP is oblique when governed by P and certain marked verbs;
- b. NP is objective when governed by V;
- c. NP is nominative when governed by Tense

(69) \( \alpha \) is governed by \( \beta \) if \( \alpha \) is c-commanded by \( \beta \) and no major category or major category boundary appears between \( \alpha \) and \( \beta \)

(70) \( \star \)N, where N has no Case

(Chomsky, 1980, 25)

Implementing case-checking as movement-to-specifier essentially asserts that there must exist some point in the derivation at which the check-er selects (a subtree containing) the check-ee as a complement. This is a precondition for the movement that does the checking. The MG selection rules implement a kind of complementation that respects Chomsky’s definition of c-command in (69). Indeed the essence of Chomsky’s surface filter (70) is that, since all candidate surface structures are generated in some way, there must exist some point in the derivation at which Case Assignment (68) applied. The primary difference between the two accounts is that in the MG, the occasion on which case-checking happens is more closely restricted. If a lexical entry is \( \gamma + \text{case} \delta \) then case must be assigned after
the elimination of the $\gamma$ but before elimination of any $\delta$. A grammar-writer might choose to associate semantic differences of other consequences with this ordering.

Fundamentally, the treatment of dative shift follows (Baker, 1988, 286). The idea is that the dative shifted version (32) and unshifted version (31)

(31) John told the story to the sailor

(32) John told the sailor the story

are identical except for two lexical facts:

the preposition in the unshifted version is overt  
\[[\text{to}]: [\text{d, +case}, 'Pto']\].

the preposition in the shifted version is silent  
\[[\emptyset]: [\text{d, +case}, 'Pto', -\text{dat}]\].

Following Oehrle (1975) particular verbs can optionally license a dative movement (+\text{dat}) that reorders the two verbal arguments. Such a lexicalist arrangement acknowledges the possibility of verb-specific exceptions to the dative alternation (e.g. \textit{donate}).

To implement the idea of prepositions checking case by movement to specifier a system of PP shells is adopted. This system posits a “little p” category analogous to “little v.” For instance, in 33 the preposition \textit{to} selects a DP.

(33) \[[\text{PP} \text{to+case} [\text{DP} \text{-case} \text{ the sailor }]]\]

(34) \[[\text{PP} [\text{PP} \text{ the sailor } ] \text{ to } ]\]

(35) \[[\text{PP} \text{ to+little p} [\text{PP} [\text{DP} \text{ the sailor } ] \text{ t}\text{o} ]\]]

Then in 34 the DP \textit{the sailor} checks its case feature by moving to the specifier of the P \textit{to}. Finally, little p takes 34 as complement, and incorporates the head \textit{to} on the left side of the silent little p, as shown in 35. Essentially, little p gets the yield back in the correct linear order following case checking. Probably there exist languages where a deep similarity between the P and V systems is more visible, although I am unaware of any relevant examples. More complete structural descriptions for unshifted (4.5) and shifted (4.6) examples appear below.

### 4.5 Tense and auxiliary verbs

The English auxiliary verb system has been a source of constant fascination for generative grammarians. At one point a flagship analysis in transformational grammar (Chomsky, 1965, 39)\footnote{See also Kimball (1973) §4.4.3 for a concise explanation of how an evaluation metric based on number of symbols might value the transformational analysis of English Auxiliaries most highly.} it later became a key battleground in the debate over the rewriting-rule-theoretic complexity of natural language (Gazdar, Pullum, and Sag, 1982). A commonly held view nowadays within computational linguistics is that English auxiliaries can be comfortably analyzed using finite-state methods (Kornai, 1982).
Following Stabler (2002) the Larsonian grammar analyzes auxiliaries using different selection feature types. Basically, there is little v and there is tense, and between these two categories is a sequence of all-optional categories Modal, Have and Be.

% auxiliary verbs
[will]::=[=‘Have’,’Modal’]. [will]::=[=‘Be’,’Modal’].
[will]::=[=little_v,’Modal’]. [have]::=[=‘Been’,’Have’].
[have]::=[=ven,’Have’]. [be]::=[=ving,’Be’].
[been]::=[=ving,’Been’].

This analysis echos Chomsky’s (1957) phrase structure rule

\[ \text{Aux} \to C(M)(\text{have} + \text{en})(be + \text{ing})(be + en). \]

However, the Larsonian grammar breaks with Chomsky by situating the aspect-related suffixes -\text{ing} and -\text{en} in little v and incorporating the lexical verb on their left side, rather than using an MG rule of affix-hopping to achieve rightward movement.

\[ ::=[\Rightarrow v,=d,little_v]. \] % it’s all optional
\[ ['-\text{en}']::=[\Rightarrow v,=d,ven]. \% PERFECT en is the symbol for past participle
\[ ['-\text{ing}']::=[\Rightarrow v,=d,ving]. \% PROGRESSIVE ing for present participle

The same technique is used for tense, which also case-checks the subject. An example exercising these categories is given in figure 4.7.

### 4.6 Modifiers

Adjunction is an experimental MG rule (Stabler, 2002) that can implement optional modification. In the version of the formalism used for the Larsonian grammar, adjunction can apply if the appropriate adjunction axiom exists in the grammar. An adjunction axiom states that a certain, indicated category can adjoin (be adjoined) on the left (on the right) to a phrase of a given second category. For instance, adjectives can left-join onto nouns.

\[ ['\text{A'}] >> [n]. \]

Since, in this version of the MG formalism, all linguistic objects are categorized (tuples of) strings, there is no principled difference between adjoining \textit{strange} to a word like \textit{girl} and adjoining \textit{yesterday} to a phrase like \textit{buy a book}. The latter is depicted in figure 4.8.

The difference between argument combination and adjunction combination is recorded in an MG derivation. Notating this difference on each derivation tree node allows the MG grammar writer to take a position on the argument/adjunct distinction, which has been
called “probably the most unclear issue in linguistics” (Aldezabal et al., 2002). The distinction is between phrases whose co-occurrence with a given verb is best recorded in the verb’s lexical entry – versus all other phrases. More precise characterization of the difference between these two classes of VP subconstituents is a challenging project. Schütze (1995) recounts some syntactic tests for adjunctionhood, drawing on earlier compendia of diagnostics (Pollard and Sag, 1987, 5.6), pointing out the significant limitations on these tests.

The only adjoined expressions in the Larsonian grammar are degree intensifiers like very and always, adjectives like young and important, temporal modifiers like yesterday and today and preposition-headed oblique verb modifiers such as with the girl and on the boat.

4.7 Grammatical relations

Since the Larsonian grammar is designed to cover a set of stimuli that systematically vary the grammatical relation of an extracted element, it necessarily takes a position on the proper notation of those grammatical relations. It adopts Chomsky’s (1965) position that grammatical relations are definable as phrase structure configurations.

It is necessary only to make explicit the relational character of these notions by defining ‘Subject-of,’ for English, as the relation holding between the NP of a sentence of the form NP \( \sim \) Aux \( \sim \) VP and the whole sentence, ‘Object-of’ as the relation between the NP of a VP of the form V \( \sim \) NP and the whole VP, etc

(Chomsky, 1965, 69)

This position has been questioned by linguists working in the Relational Grammar, LFG, and subsequent traditions.

It is not clear whether or not the type of configurational definitions of grammatical relations in Chomsky 1965 can be extended to derived structures, even for English. One of the consequences of the Relational Succession Law – the recognition of the necessity of grammatical relations at all levels – was an important factor that led to making grammatical relations independent of positions in phrase structure configurations. The result was to abandon attempts to define grammatical relations in terms of other notions and to take them to be primitives of grammatical theory. This marked the beginning of relational grammar.

(Perlmutter and Postal, 1983, 59)

Particular definitions of grammatical relations in the Larsonian grammar are given in figure 4.9.

Baker (2001) provides an enlightening status report on the primitivity of grammatical relations debate. While acknowledging that constituency data (on which the phrase-structural
account of grammatical relations is supposedly based) is often weak in non-English languages, he points out that the problem has been made only worse by the explosion of proposals for silent functional projections. “If one accepts these kinds of phrase structures, then one is clearly forced to acknowledge that not all phrases can be revealed by traditional phrase structure tests in any simple way,” he writes. Continuing, “one cannot cleft a small clause, the NP complement of a determiner, or a Larsonian shell,” (page 28) Baker reiterates the Chomskyan view of phrase structure as more indirectly connected to the data than are a set of immediate constituency assertions. He challenges RG proponents to substantiate their criteria for subjecthood, objecthood, chômeurhood and all RG theoretical primitives. The explanatory benefits of phrase structure for explaining control, coreference and passivization facts, he suggests, should be retained and bolstered by a more accurate theory of what constituency tests actually measure.
4.8 Appendix: the Larsonian grammar itself

% file: larsonian.pl
% author: John Hale
% created: Thu Jul 18 11:46:02 EDT 2002
% updated: Sun Jul 21 15:40:05 EDT 2002
% updated: Thu Aug 22 12:05:52 EDT 2002
%
% make preposition and tense case assigners,
% require the dp’s move to "check" case
% as was done in Stabler’s g6.pl
%
% treat dative alternation as in Baker’s UTAH chapter in
% Haegeman 1997 "Elements of Grammar" i.e. in dative,
% preposition is empty and the dative argument moves to get case.
%
% note that Ura’s proposal is backwards from Larson’s 88 account
% while Ura assumes the IO must move to get case, Larson construes
% ’to as "Case-marking" (section 5.1). adopt Larsonian arrangement.
%
% retain, as always, the Kaynian treatment of RCs

[] ::= [=t,c]. % regular empty complementizer
[] ::= [=t,\text{wh_rel,c_rel}]. % wh-hoisting complementizer
[that] ::= [=t,'Ce']. % embedding complementizer

[the] ::= [=c_rel,d,-case]. % relative determiner
[the] ::= [=\text{Num}',d,-case]. % ordinary determiner
[a] ::= [=\text{Num}',d,-case].
[my] ::= [=\text{Num}',d,-case]. % genitive determiner
[one] ::= [=\text{Num}',d,-case]. % numerical determiner

% number
[] ::= [=n,'Num']. [%'-s'] ::= [n=>,'Num'].

% proper noun
[‘John’] ::= [d,-case]. [‘Chris’] ::= [d,-case].
[‘Dick’] ::= [d,-case]. [‘David’] ::= [d,-case]. [‘Penny’] ::= [d,-case].
[‘Sally’] ::= [d,-case]. [‘Paul’] ::= [d,-case]. [‘Stephen’] ::= [d,-case].
[‘Mary’] ::= [d,-case]. [‘Pat’] ::= [d,-case]. [‘Sue’] ::= [d,-case].
[‘Joe’] ::= [d,-case]. [‘Patrick’] ::= [d,-case]. [‘Jim’] ::= [d,-case].
[‘Jenny’] ::= [d,-case]. [‘Clare’] ::= [d,-case]. [‘Ann’] ::= [d,-case].

% indefinites, nominalization
[reading] :: [d,-case].

% pronouns
[us] ::= [d,-case].
% possessive
% should be the complement of an external determiner
% in order to be pied-piped during wh-promotion
[%s?]:=[='Num',='Num',n].

[which]:=[='Num',+f,d,-case,-wh_rel].  % 'promoting' wh-words
[who]:=[='Num',+f,d,-case,-wh_rel].  % semantic selection features

% common nouns
[story]:[n].  [story]:[n,-f].  % w and w/o promotion feature
[boy]:[n].  [boy]:[n,-f].
[girl]:[n].  [girl]:[n,-f].
[man]:[n].  [man]:[n,-f].
[woman]:[n].  [woman]:[n,-f].
[dog]:[n].  [dog]:[n,-f].
[cat]:[n].  [cat]:[n,-f].
[book]:[n].  [book]:[n,-f].
[house]:[n].  [house]:[n,-f].
[town]:[n].  [town]:[n,-f].
[trick]:[n].  [trick]:[n,-f].
[treat]:[n].  [treat]:[n,-f].
[lie]:[n].  [lie]:[n,-f].
[sweet]:[n].  [sweet]:[n,-f].
[present]:[n].  [present]:[n,-f].
[ticket]:[n].  [ticket]:[n,-f].
[letter]:[n].  [letter]:[n,-f].
[picture]:[n].  [picture]:[n,-f].
[answer]:[n].  [answer]:[n,-f].
[accident]:[n].  [accident]:[n,-f].
[box]:[n].  [box]:[n,-f].
[apple]:[n].  [apple]:[n,-f].
[food]:[n].  [food]:[n,-f].
[cake]:[n].  [cake]:[n,-f].
[ship]:[n].  [ship]:[n,-f].
[car]:[n].  [car]:[n,-f].
[sailor]:[n].  [sailor]:[n,-f].
[uncle]:[n].  [uncle]:[n,-f].
[mother]:[n].  [mother]:[n,-f].
[father]:[n].  [father]:[n,-f].
[brother]:[n].  [brother]:[n,-f].
[friend]:[n].  [friend]:[n,-f].
[bill]:[n].  [bill]:[n,-f].
[leg]:[n].  [leg]:[n,-f].
[clothe]:[n].  [clothe]:[n,-f].

% ....with complements
[fact]:=[='Cs',='n'].

% preposition is a case assigner (Haegeman p193)
[to]: [=d,+case,'Pto']. [on]: [=d,+case,'Pon'].
[for]: [=d,+case,'Pfor']. [with]: [=d,+case,'Pwith'].
[in]: [=d,+case,'Pin'].
[]: [=d,+case,'Pto',-dat]. % P is empty in dative

% little p
[]: [=>'Pto',p_to]. []: [=>'Pin',p_in]. []: [=>'Pwith',p_with].
[]: [=>'Pfor',p_for].

ditransitive verbs - dative alternating
[tell]: [=p_to,=d,+case,+dat,v]. [give]: [=p_to,=d,+case,+dat,v].
[show]: [=p_to,=d,+case,+dat,v]. [explain]: [=p_to,=d,+case,+dat,v].
[teach]: [=p_to,=d,+case,+dat,v]. [sell]: [=p_to,=d,+case,+dat,v].

ditransitive verbs - not
[tell]: [=p_to,=d,+case,v]. [give]: [=p_to,=d,+case,v].
[show]: [=p_to,=d,+case,v].
[explain]: [=p_to,=d,+case,v]. [teach]: [=p_to,=d,+case,v].
[sell]: [=p_to,=d,+case,v].

% intransitive verbs
[matter]: [v]. [wait]: [v]. [rule]: [v]. % yeah...yeah...like Slayer!

transitive verbs
[tell]: [=d,+case,v]. [love]: [=d,+case,v]. [hate]: [=d,+case,v].
[bring]: [=d,+case,v]. [take]: [=d,+case,v]. [send]: [=d,+case,v].
[have]: [=d,+case,v]. [surprise]: [=d,+case,v].
[pay]: [=p_for,v]. % pay...for services
[pay]: [=d,+case,v]. % pay...the piper
[come]: [=‘A’,v]. % came late

% CP-taking verbs
[know]: [=‘Ce’,v]. [forget]: [=‘Ce’,v]. [remember]: [=‘Ce’,v].

% auxiliary verbs
[will]: [=‘Have’,‘Modal’]. [will]: [=‘Be’,‘Modal’].
[will]: [=‘little_v’,‘Modal’]. [have]: [=‘Been’,‘Have’].
[have]: [=‘ven’,‘Have’]. [be]: [=‘ving’,‘Be’].
[been]: [=‘ving’,‘Been’].

% little v gets the subject
[]: [=v,=d,little_v]. % it’s all optional
[‘-en’]: [=v,=d,ven]. % PERFECT en is the symbol for past participle
% Lester, p64: next most common past participle ending
% is vowel change along,
% e.g. begin-began-begun, sink-sank-sunk etc
% However, for many irregular verbs and for all regular verbs
% the form of the past participle is identical with the form
% of the simple past
% e.g. tell-told-told, leave-left-left play-played-played,
% talk-talked-talked so the string
%  have -en talk
%  have talk -en
%  by morphological rules
%  have talked
% likewise, 'have -en swim' derives 'have swum'

[‘-ing’]:=[>v,=d,ving].% PROGRESSIVE ing is the symbol for
% present participle

% tense
[] :=[>little_v,+case,t]. [‘-s’]:=[>little_v,+case,t].
[‘-ed’]:=[>little_v,+case,t].
[] :=[>Modal’,+case,t]. [‘-s’]:=[>Modal’,+case,t].
[‘-ed’]:=[>Modal’,+case,t].
[] :=[>Have’,+case,t]. [‘-s’]:=[>Have’,+case,t].
[‘-ed’]:=[>Have’,+case,t].
[] :=[>Be’,+case,t]. [‘-s’]:=[>Be’,+case,t].
[‘-ed’]:=[>Be’,+case,t].

% ignore do-support and negation for now
[‘doesn’t’]:=[little_v,+case,t].

% predicative/copular be
[be] :=[a,’Be’].

% little a
[] :=[‘A’,=d,a].
[] :=[d,+case,=d,a]. % [ADJ the bomb] [ADJ a dear]

% adjectives
[young]:=[‘A’]. [poor]:=[‘A’]. [clever]:=[‘A’]. [gentle]:=[‘A’].
[kind]:=[‘A’]. [proud]:=[‘A’]. [lost]:=[‘A’]. [cheap]:=[‘A’].
[interesting]:=[‘A’]. [sad]:=[‘A’]. [late]:=[‘A’]. [ill]:=[‘A’].
[important]:=[‘A’]. [angry]:=[‘A’]. [pretty]:=[‘A’]. [honest]:=[‘A’].
[right]:=[‘A’]. [strange]:=[‘A’]. [old]:=[‘A’]. [long]:=[‘A’].

% optional intensifiers
[so]:=[deg]. [very]:=[deg]. [always]:=[deg]. [too]:=[deg].

% which can left-adjoin to adjectives
[deg]>[‘A’].

% adjectives can also left-adjoin onto nouns
[‘A’]>=[n].
% temporal modifiers
[yesterday]::[tmp]. [today]::[tmp].

% which right adjoin to verbs
[v]@[tmp].

% oblique modifiers can right adjoin to VPs

startCategory(c).
subject
they had forgotten that the boy who
told the story was so young
the fact that the girl who paid for the
tickets is very poor doesn’t matter
I know that the girl who got the right
answer is clever
he remembered that the man who sold
the house left the town

oblique
they had forgotten that the box which
Pat brought with apples in was lost
the fact that the girl who Sue wrote
the story with is proud doesn’t matter
I know that the ship which my uncle
took Joe on was interesting
he remembered that the food which
Chris paid the bill for was cheap

direct object
they had forgotten that the letter
which Dick wrote yesterday was so long
the fact that the cat which David
showed to the man likes eggs is strange
I know that the dog which Penny
bought today is very gentle
he remembered that the sweets which
David gave Sally were a treat

genitive subject
they had forgotten that the girl whose
friend bought the cake was waiting
the fact that the boy whose brother
tells lies is always honest surprised us
I know that the boy whose father sold
the dog is very sad
he remembered that the girl whose
mother sent the clothes came too late

indirect object
they had forgotten that the man who
Ann gave the present to was old
the fact that the boy who Paul sold
the book to hates reading is strange
I know that the man who Stephen ex-
plained the accident to is kind
he remembered that the dog which
Mary taught the trick to was clever

genitive object
they had forgotten that the man
whose house Patrick bought was so ill
the fact that the sailor whose ship Jim
took had one leg is important
I know that the woman whose car
Jenny sold was very angry
he remembered that the girl whose
picture Clare showed us was pretty

Figure 4.1: Stimuli from Hawkins and Keenan 1974/1987

Figure 4.2: IO merged first, then DO, then Subject
Figure 4.3: Simplified promotion analysis of an object-extracted relative clause

Figure 4.4: extraction from genitive subject: who + 's → whose
Figure 4.5: unshifted dative

Figure 4.6: shifted dative
Figure 4.7: Some auxiliaries

Figure 4.8: VP modifier
<table>
<thead>
<tr>
<th>grammatical relation</th>
<th>configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>complement of little v</td>
</tr>
<tr>
<td>direct object</td>
<td>complement of a transitive verb or specifier of a ditransitive verb</td>
</tr>
<tr>
<td>indirect object</td>
<td>complement of ditransitive verb</td>
</tr>
<tr>
<td>oblique</td>
<td>prepositional element right-adjointed to VP</td>
</tr>
<tr>
<td>genitive subject</td>
<td>complement of little v that is itself a complement of possessive marker</td>
</tr>
<tr>
<td>genitive object</td>
<td>complement of transitive verb or specifier of ditransitive verb that is itself complement of a transitive verb or specifier of a ditransitive verb</td>
</tr>
</tbody>
</table>

Figure 4.9: configurational definitions of grammatical relations
Chapter 5

Another Look at the Accessibility Hierarchy

The empirical basis of chapter 3 was the Accessibility Hierarchy (AH) of Keenan and Comrie (1977) which is depicted below.

SUBJECT ⊆ DIRECT OBJECT ⊆ INDIRECT OBJECT ⊆ OBLIQUE ⊆ GENITIVE ⊆ OCOMP

The AH is an implicational markedness hierarchy; if, for example, a language has a rule for forming OBLIQUE relative clauses, it necessarily also has rules for forming INDIRECT OBJECT, DIRECT OBJECT and SUBJECT relative clauses. The if-then nature of these subset relationships motivates the term “implicational” markedness hierarchy. Although the AH was originally proposed as a cross-linguistic typological generalization, chapter 3 built on the work of Keenan and Hawkins (1987) by proposing a theoretical explanation of the AH viewed as a processing phenomenon. The processing phenomenon Keenan and S. Hawkins found was a correlation between repetition errors and position on the AH. The theoretical explanation was that, on a realistic probabilistic grammar, more information is transacted between speaker and hearer in sentences farther out on the hierarchy. Being under increasing cognitive load during comprehension, subjects are more likely to make mistakes repeating back these sorts of sentences.

However satisfying this explanation may be, chapter 3 also turned up certain quirks of the Keenan and Hawkins (1987) stimuli. For instance, the low frequency of plural noun phrases in the stimulus set led to outliers in the predictions that went away when a uniform training set was used. Indeed, the naturalistic character of the stimuli, which (although matched for word frequency) exhibit a variety of syntactic constructions, a variety of verb subcategorizations and an erratic rate of optional modification, may in fact obscure rather than elucidate the underlying processing phenomenon.

Beyond these considerations of materials, there are also methodological questions.
Chapter 3 tested a theory of word-by-word reading difficulty on whole-sentence repetition accuracy scores. If this theory is useful, it will be in providing a guide to the world of sentence processing at the resolution of a morpheme-in-a-sentence. In order to make this determination of utility, it is necessary to find out what the empirical situation actually is. While it is known that subject-extracted relative clauses are easier to read than object-extracted relative clauses, on a variety of behavioral measures (reading time: (King and Just, 1991) phoneme monitoring; (Hakes, Evans, and Brannon, 1976) continuous lexical decision: (Ford, 1983) comprehension question accuracy: (Holmes and O'Regan, 1981) list recall accuracy: (Wanner and Maratsos, 1978)), the work of Keenan and Hawkins (1987) implicates grammatical relations as a properly linguistic determinant of sentence processing difficulty.

The three experiments reported in this chapter examine individual implications of the (processing extension of) the AH. The first experiment examines the role of genitivity in the processing of relative clauses. Does a relative clause's being extracted from a genitive context lead to increased processing difficulty over nongenitive subject or object-extracted relative clauses, as predicted by the AH? The second looks at the directness of object-extraction. If relativization is from indirect object, as opposed to direct object, is comprehension more difficult? The third and final experiment examines obliqueness. Experiment 3 compares the comprehension difficulty of relative clauses extracted from oblique as opposed to other grammatical relations.

To preview the results, experiment 1 finds that genitivity does affect comprehension difficulty as one would expect given the AH. Since an information processing account of slightly different genitive stimuli was presented in chapter 3, (see figures 3-24 and 3-25) the present chapter simply remarks that the empirical finding confirms the AH (and its reduct to information theory and generative grammar).

Experiment 2, while sensitive enough to replicate the well-known subject/object asymmetry, does not find a corresponding direct/indirect object asymmetry, contra the AH.

Finally, experiment 3 confirms the AH's prediction that oblique-extracted relative clause processing is harder than direct object-extract relative clause processing. This asymmetry appears at the main verb, rather than on embedded verb where the subject/object asymmetry is typically observed. Some speculations on possible explanations for this result are provided.

5.1 Experiment 1: genitives

Experiment 1 explores the role of genitive modification on extracted NPs. It is a 2 X 2 experiment crossing the grammatical relation extracted-from (subject or object) with
the presence or absence of genitive modification. Stimuli are of the form:

<table>
<thead>
<tr>
<th>type</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU</td>
<td>The hairdresser’s daughter, who insulted the beautician’s sister, got in an accident.</td>
</tr>
<tr>
<td>DO</td>
<td>The beautician’s sister, who the hairdresser’s daughter insulted, got in an accident.</td>
</tr>
<tr>
<td>GenS</td>
<td>The hairdresser, whose daughter insulted the beautician’s sister, got in an accident.</td>
</tr>
<tr>
<td>GenO</td>
<td>The beautician, whose sister the hairdresser’s daughter insulted, got in an accident.</td>
</tr>
</tbody>
</table>

According to the AH, conditions SU and GenS, being subject-extracted relative clauses, should be easier than conditions DO and GenO, which are object-extracted. This follows because subject precedes object on the hierarchy. Likewise, since genitive comes after both subject and object on the hierarchy, the AH predicts that conditions GenS and GenO should be harder than SU and DO since they are extracted from within a genitive noun phrase. By contrast, SU and DO extract an entire noun phrase that just happens to have genitive internal structure.

5.1.1 Method

Participants

Forty-eight participants from the MIT community participated. They were compensated with nominal payment. All were native speakers of English and were naive as to the purposes of the study.

Materials

Twenty-four sets of stimuli were constructed\(^1\). The subject noun phases of each member was viewed as being “derived” from a non-relativized canonical sentence such as:

The hairdresser’s daughter insulted the beautician’s sister.

Relativizing the subject of this canonical sentence results in condition SU, relativizing the possessor-object results in condition GenO, et cetera. This improves on the Keenan and Hawkins (1987) stimuli because it keeps the meaning of each sentence approximately constant, using essentially the same words in a different order. In each case, care was taken to ensure that each noun phrase is equally plausible as subject or object of the embedded verb e.g. insulted. The same trailing verb phrase ...got in an accident was used in all conditions. A full list of stimuli can be found in appendix 1. 64 stimuli from an unrelated experiment, as well as 52 other, syntactically-diverse sentences served as fillers.

\(^1\)The author thanks Amy Daitch for her hard work constructing stimuli for all three experiments.
Procedure

Participants read stimulus sentences word by word using a moving window display (Just, Carpenter, and Woolley, 1982). On a computer screen, the words of the sentence were replaced with hyphens. As the participants pressed the spacebar, successive words were revealed, and then converted back to hyphens at the next spacebar press. The time between these keypresses was stored. After each sentence, to promote natural sentence-understanding, there was a Yes/No comprehension question (see appendix 1). Before the experiment as such got underway, participants were familiarized with self-paced reading through several practice items. Participants were instructed to read naturally ensuring full comprehension.

5.1.2 Results

Data from nine subjects whose accuracy on the comprehension questions was less than 66%, were excluded. From the remaining data, only trials on which participants correctly answered the comprehension question were considered.

Analyses of variance were performed on the reading times measured at various points in the stimulus sentences, trimming data to within 3 standard deviations from subjects’ mean reading time per-word. Results are reported both by subjects ($F_1$) and by items ($F_2$).

Previous work (Grodner, Watson, and Gibson, 2000) localized the difference between subject and object relative clauses on the embedded verb. At this point (word 5 in the subject relatives, word 8 in the object relatives) there are main effects of both genitivity $F_1(1,38) = 4.309, p < 0.05$, $F_2(1,23) = 8.284, p < 0.01$ and subjecthood $F_1(1,38) = 27.518, p < 0.001$, $F_2(1,23) = 83.299, p < 0.001$ of the relativized noun phrase. There was no interaction between genitivity and subjecthood at the embedded verb. No effect of either genitivity or subjecthood was observed at the main verb (words 9 and 10). In the initial wh-word/noun-pair (words 2-4), the genitives are read significantly faster than the nongenitives $F_1(1,38) = 43.617, p < 0.0001$, $F_2(1,23) = 16.444, p < 0.0001$. To de-confound end-of-embedded clause and embedded verb, an analysis of variance was conducted on the last four words of the embedded clause (words 5-8). This larger region holds constant the number of verbs and nouns across conditions. Here, main effects of both genitivity $F_1(1,38) = 21.610, p < 0.0001$, $F_2(1,23) = 21.908, p < 0.0001$ and subjecthood $F_1(1,38) = 21.882, p < 0.0001$, $F_2(1,23) = 9.321, p < 0.01$ were also observed. There was an effect of genitivity but not subjecthood, on question-answering accuracy, $F_1(1,38) = 6.673, p < 0.05$. $F_2(1,23) = 7.929, p < 0.01$.

Average word-by-word reading times are shown in figure 5.1. Table 5.1 summarizes
the results.

![Experiment 1 graph](image)

**Figure 5.1:** Experiment 1 average word-by-word reading times

<table>
<thead>
<tr>
<th>condition</th>
<th>genitivity</th>
<th>subjecthood</th>
<th>mean RT (msec) words 5-8</th>
<th>question-answering accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU</td>
<td>nongen</td>
<td>subj</td>
<td>474.328</td>
<td>87</td>
</tr>
<tr>
<td>DO</td>
<td>nongen</td>
<td>obj</td>
<td>576.313</td>
<td>85</td>
</tr>
<tr>
<td>GenS</td>
<td>gen</td>
<td>subj</td>
<td>577.314</td>
<td>79</td>
</tr>
<tr>
<td>GenO</td>
<td>gen</td>
<td>obj</td>
<td>721.008</td>
<td>77</td>
</tr>
</tbody>
</table>

**Table 5.1:** Results of Experiment 1

### 5.1.3 Discussion

Replicating the previous work discussed above, a subject/object asymmetry was found in the reading times at the embedded verb. What is novel is the demonstration of a corresponding genitive/non-genitive asymmetry. In both subject- and object- extracted relative clauses, if the extraction is also from genitive, it is read more slowly. This confirms the prediction of the AH that genitivity makes relative clause processing harder. If adjacent points on the AH are taken to be equally distant from one another across the hierarchy, then the AH also predicts the observed lack of interaction between subjecthood and genitivity. Contra the AH, nongenitive object extraction is not read significantly faster than genitive subject extraction.
5.2 Experiment 2: direct versus indirect objects

Experiment 2 compares extraction from different argument positions of ditransitive verbs like *give*. The stimuli are of the form:

<table>
<thead>
<tr>
<th>type</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU</td>
<td>The secretary who sent the student to the administrator talked to the librarian.</td>
</tr>
<tr>
<td>DO</td>
<td>The student who the secretary sent to the administrator talked to the librarian.</td>
</tr>
<tr>
<td>IO</td>
<td>The administrator to whom the secretary sent the student talked to the librarian.</td>
</tr>
</tbody>
</table>

Previous work (including experiment 1) predicts greater reading time on the DO type than on the SU type; the “subject/object asymmetry.” In addition, the AH predicts a “direct/indirect object asymmetry.” If indirect object-extracted relative clauses are harder to process than direct object-extracted relative clauses, in virtue of being farther down the hierarchy, then reading times in the IO condition should be the highest.

5.2.1 Method

The methods of experiment 2 were exactly the same as in experiment 1. Trials were interleaved such that the same 48 people who were run on experiment 1 also participated in experiment 2. A full list of stimuli can be found in appendix 2.

5.2.2 Results

Data from the same nine subjects whose comprehension question-answering accuracy was lower than 66% were again excluded. Only questions on which participants correctly answered the comprehension question were considered, and reading times were again trimmed to within 3 standard deviations from subjects’ mean reading time per-word. Results are reported both by subjects ($F_1$) and by items ($F_2$).

Replicating subject/object asymmetry found in experiment 1, reading times at the embedded verb (e.g., *sent* and subsequent function word) were significantly faster in the SU condition than in the DO condition $F_1(1, 38) = 20.161, p < 0.001$, $F_2(1, 17) = 13.649, p < 0.005$.

By contrast, the reading time difference between the DO and IO condition at the embedded verb (and subsequent word) did not reach statistical significance $F_1(1, 38) = 2.935, p = 0.095$, $F_2(1, 17) = 1.724, p = 0.207$.

Nor was the reading times difference between DO and IO conditions or between SU and DO conditions statistically significantly at the main verb (words 10 and 11) or in the region following the WH-word, up to but not including the main verb (words 4-9, or 5-9 in the IO condition).
There was a main effect of AH position on question-answering accuracy, $F_1(2, 76) = 9.676, p < 0.001, F_2(2, 34) = 4.864, p < 0.05$. Question-answering accuracy was significantly less accurate in the IO than in the DO condition by subjects $F_1(1, 38) = 9.561, p < 0.005$, but not by items $F_2(1, 17) = 3.984, p = 0.062$.

Average word-by-word reading times are shown in figure 5.2. The results are summarized in table 5.2.

![Figure 5.2: Experiment 2 average word-by-word reading times](image)

<table>
<thead>
<tr>
<th>relation</th>
<th>mean RT (msec) at embedded V + next word</th>
<th>question-answering accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU</td>
<td>368.631</td>
<td>89</td>
</tr>
<tr>
<td>DO</td>
<td>457.656</td>
<td>86</td>
</tr>
<tr>
<td>IO</td>
<td>430.208</td>
<td>74</td>
</tr>
</tbody>
</table>

Table 5.2: Results of Experiment 2

5.2.3 Discussion

While experiment 2 was sensitive enough to replicate the subject/object asymmetry, it did not find a corresponding direct-object/indirect-object asymmetry as predicted by the AH. It did find an effect of the AH in the question-accuracy data. In particular subjects’ reading, although about equally slow in the DO and IO conditions, is significantly less accurate in the IO condition. This decreasing question-answering accuracy is consistent with the repetition-accuracy findings of Keenan and Hawkins (1987).
5.3 Experiment 3: obliques

Experiment 3 examines extraction from benefactive oblique arguments, comparing them with extraction from subject and object arguments of the same verb. Stimuli are of the form

<table>
<thead>
<tr>
<th>type</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU</td>
<td>The officer who pacified the captor for the hostage held a knife.</td>
</tr>
<tr>
<td>DO</td>
<td>The captor who the officer pacified for the hostage held a knife.</td>
</tr>
<tr>
<td>OBL</td>
<td>The hostage for whom the officer pacified the captor held a knife.</td>
</tr>
</tbody>
</table>

According to the AH, oblique extraction should be harder than either subject- or object-extraction. If this is so, reading times should be higher in the OBL condition than in any other.

5.3.1 Method

The methods of experiment 3 were exactly the same as in experiments 1 and 2. Thirty-nine members of the MIT community participated. A full list of stimuli can be found in appendix 3.

5.3.2 Results

Data from seven subjects who scored lower than 66% on the comprehension questions was discarded. Analyses of variance were again performed on the reading times at the embedded verb (e.g. *pacified*) as well as the main verb (e.g. *held*) and only trials on which participants correctly answered the comprehension question were considered. Reading times were again trimmed to within 3 standard deviations from subjects’ mean reading time per-word. Results are reported both by subjects ($F_1$) and by items ($F_2$).

Replicating the results of experiment 1, at the embedded verb (e.g. *pacified* and successive word) a subject/direct object asymmetry was observed $F_1(1, 32) = 17.690, p < 0.0001, F_2(1, 17) = 44.280, p < 0.0001$. However, a comparable direct object/oblique asymmetry was not observed there $F_1(1, 32) = 0.364, p = 0.550, F_2(1, 17) = 0.034, p = 0.856$. Instead, at the main verb (e.g. *held* and successive word), readings times in OBL were significantly slower than in DO $F_1(1, 32) = 15.005, p < 0.0001, F_2(1, 17) = 4.879, p < 0.05$.

The region immediately following the WH-word, but before the main verb (the region combining enverb and phrase3) holds constant the number of verbs and nouns. In this region, reading times are significantly different in DO and OBL conditions by subjects $F_1(1, 32) = 5.176, p < 0.05$, but not by items $F_2(1, 17) = 2.998, p = 0.101$. 
A main effect of AH position was observed in subjects’ comprehension question accuracy $F_1(2, 64) = 6.562, p < 0.01$, $F_2(2, 34) = 3.529, p < 0.05$.

Average regional reading times are shown in figure 5.3. The results are summarized in table 5.3. This table includes the standard deviations in parenthesis to highlight the observation of unusually high variance in the reading times at the main verb in the OBL condition.

<table>
<thead>
<tr>
<th>relation</th>
<th>mean RT (msec) at embedded V + next word</th>
<th>RT at main verb + question answering accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU</td>
<td>332.441 (sd=74.0513)</td>
<td>376.478 (sd=150.369) 88</td>
</tr>
<tr>
<td>DO</td>
<td>423.345 (sd=162.195)</td>
<td>397.112 (sd=148.859) 82</td>
</tr>
<tr>
<td>OBL</td>
<td>436.810 (sd=174.732)</td>
<td>472.860 (sd=202.565) 75</td>
</tr>
</tbody>
</table>

Table 5.3: Results of Experiment 3

![Figure 5.3: Experiment 3 average regional reading times](image)

5.3.3 Discussion

Experiment 3 again replicated the subject-object asymmetry, and found an direct-object/oblique asymmetry at the main verb (e.g. *held*) but not at the embedded verb (e.g. *pacified*). This is to say that obliques are just as hard as extractions from direct object at their embedded verbs, but obliques become harder by the time the reader processes the main verb. A tantalizing possibility is that the processing difficulty encountered in extrac-
tion from oblique is qualitatively different from subject- and object-extracted relatives. Of course, this suggestion is qualified by the very high variance observed in main verb reading times.

Consistent with the results of experiments 1, 2, and the findings of Keenan and Hawkins (1987), subjects were significantly less accurate answering questions about relative clauses formed on positions farther out on the AH, such as obliques.

5.4 General Discussion

In some respects, the AH as a processing hypothesis has been upheld. Experiment 1 found both a subject/object-extraction asymmetry, and a genitive/nongenitive asymmetry, both of which are predicted by the AH. Experiment 3 found a direct-object/oblique asymmetry at main verbs. In others, the processing extension of the AH has been disconfirmed. Experiment 1 did not find support for the DIRECT OBJECT→GENITIVE SUBJECT implication, and experiment 2 failed to find a direct-object/indirect-object asymmetry.

5.5 Conclusion

Further empirical work is necessary to accurately chart the processing characteristics of the sentence types the AH classifies. As a processing hypothesis, the AH is only partially correct. For the phenomena that it does correctly predict, even more specific explanations, in terms of the entropy reduction hypothesis 1, are at hand.
Appendix 1: experiment 1 stimuli

(setq fill-column 72)
# e1 1a
The hairdresser’s daughter, who insulted the beautician’s sister, got in an accident.
? Did the hairdresser insult the beautician?  N

# e1 1b
The beautician’s sister, who the hairdresser’s daughter insulted, got in an accident.
? Did the hairdresser get in an accident?  N

# e1 1c
The hairdresser, whose daughter insulted the beautician’s sister, got in an accident.
? Did the daughter insult the sister?  Y

# e1 1d
The beautician, whose sister the hairdresser’s daughter insulted, got in an accident.
? Did the beautician get in an accident?  Y

# e1 2a
The professor’s assistant, who knew the Dean’s wife, played the piano.
? Did the assistant play the piano?  Y

# e1 2b
The Dean’s wife, who the professor’s assistant knew, played the piano.
? Did the assistant know the wife?  Y

# e1 2c
The professor, whose assistant knew the Dean’s wife, played the piano.
? Did the professor play the piano?  Y

# e1 2d
The Dean, whose wife the professor’s assistant knew, played the piano.
? Did the professor know the Dean?  N

# e1 3a
The band’s groupie, who loved the guitarist’s replacement, took the subway.
? Did the groupie love the replacement?  Y

# e1 3b
The guitarist’s replacement, who the band’s groupie loved, took the subway.
? Did the guitarist take the subway?  N

# e1 3c
The band, whose groupie loved the guitarist’s replacement, took the subway.
? Did the guitarist love the groupie?  N

# e1 3 d
The guitarist, whose replacement the band’s groupie loved, took the subway.
? Did the replacement take the subway?  N

# e1 4 a
The doctor’s intern, who consulted the patient’s nurse, wrote down the information.
? Did the doctor write down the information?  N

# e1 4 b
The patient’s nurse, who the doctor’s intern consulted, wrote down the information.
? Did the nurse consult the intern?  N

# e1 4 c
The doctor, whose intern consulted the patient’s nurse, wrote down the information.
? Did the intern write down the information?  N

# e1 4 d
The patient, whose nurse the doctor’s intern consulted, wrote down the information.
? Did the intern consult the nurse?  Y

# e1 5 a
The drummer’s roommate, who punched the diva’s boyfriend, slammed the door.
? Did the drummer punch the boyfriend?  N

# e1 5 b
The diva’s boyfriend, who the drummer’s roommate punched, slammed the door.
? Did the boyfriend slam the door?  Y

# e1 5 c
The drummer, whose roommate punched the diva’s boyfriend, slammed the door.
? Did the drummer punch the boyfriend?  N

# e1 5 d
The diva, whose boyfriend the drummer’s roommate punched, slammed the door.
? Did the boyfriend slam the door?  N

# e1 6 a
The quarterback’s coach, who befriended the waterboy’s father, ran twice a day.
? Did the coach run twice a day?  Y

# e1 6 b
The waterboy’s father, who the quarterback’s coach befriended, ran twice a day.
? Did the coach befriend the father?  Y

# e1 6 c
The quarterback, whose coach befriended the waterboy’s father, ran twice a day.
? Did the quarterback run twice a day?  Y

# e1 6 d
The waterboy, whose father the quarterback’s coach befriended, ran twice a day.
? Did the coach befriend the father?  Y

# e1 7 a
The mayor’s fiancee, who invited the governor’s advisor, brought the champagne.
? Did the mayor invite the governor?  N

# e1 7 b
The governor’s advisor, who the mayor’s fiancee invited, brought the champagne.
? Did the advisor bring the champagne?  Y

# e1 7 c
The mayor, whose fiancee invited the governor’s advisor, brought the champagne.
? Did the mayor invite the advisor?  N

# e1 7 d
The governor, whose advisor the mayor’s fiancee invited, brought the champagne.
? Did the governor bring the champagne?  Y

# e1 8 a
The librarian’s helper, who angered the jeweler’s inspector, fell down the stairs.
? Did the helper fall down the stairs?  Y

# e1 8 b
The jeweler’s inspector, who the librarian’s helper angered, fell down the stairs.
? Did the inspector anger the helper?  N

# e1 8 c
The librarian, whose helper angered the jeweler’s inspector, fell down the stairs.
? Did the librarian fall down the stairs?  Y

# e1 8 d
The jeweler, whose inspector the librarian’s helper angered, fell down the stairs.
? Did the helper anger the jeweler?  N

# e1 9 a
The tutor’s student, who helped the cashier’s co-worker, bought a new puppy.
? Did the student help the cashier?  N

# e1 9 b
The cashier’s co-worker, who the tutor’s student helped, bought a new puppy.
? Did the co-worker buy a new puppy?  Y

# e1 9 c
The tutor, whose student helped the cashier’s co-worker, bought a new puppy.
? Did the student help the co-worker?  Y

# e1 9 d
The cashier, whose co-worker the tutor’s student helped, bought a new puppy.
? Did the cashier buy a new puppy?  Y

# e1 10 a
The senator’s waitress, who hugged the bartender’s customer, wore a white shirt.
? Did the bartender wear a white shirt?  N

# e1 10 b
The bartender’s customer, who the senator’s waitress hugged, wore a white shirt.
? Did the senator hug the bartender?  N

# e1 10 c
The senator, whose waitress hugged the bartender’s customer, wore a white shirt.
? Did the waitress wear a white shirt?  N

# e1 10 d
The bartender, whose customer the senator’s waitress hugged, wore a white shirt.
? Did the senator hug the bartender?  N

# e1 11 a
The lawyer’s client, who sued the engineer’s contractor, broke his finger.
? Did the client sue the contractor?  Y

# e1 11 b
The engineer’s contractor, who the lawyer’s client sued, broke his finger.
? Did the engineer break his finger?  N

# e1 11 c
The lawyer, whose client sued the engineer’s contractor, broke his finger.
? Did the client sue the engineer?  N

# e1 11 d
The engineer, whose contractor the lawyer’s client sued, broke his finger.
? Did the contractor break his finger?  N

# e1 12 a
The celebrity’s guest, who called the athlete’s masseuse, won the lottery.
? Did the guest win the lottery?  Y

# e1 12 b
The athlete’s masseuse, who the celebrity’s guest called, won the lottery.
? Did the guest call the masseuse?  Y

# e1 12 c
The celebrity, whose guest called the athlete’s masseuse, won the lottery.
? Did the celebrity win the lottery?  Y

# e1 12 d
The athlete, whose masseuse the celebrity’s guest called, won the lottery.
? Did the guest call the masseuse?  Y

# e1 13 a
The detective’s colleague, who recognized the President’s secretary, hung up the phone.
? Did the colleague recognize the secretary?  Y

# e1 13 b
The President’s secretary, who the detective’s colleague recognized, hung up the phone.
? Did the secretary hang up the phone?  Y

# e1 13 c
The detective, whose colleague recognized the President’s secretary, hung up the phone.
? Did the detective recognize the secretary?  N

# e1 13 d
The President, whose secretary the detective’s colleague recognized, hung up the phone.
? Did the President hang up the phone?  Y

# e1 14 a
The model’s photographer, who complemented the actress’ tailor, drove a Mercedes.
? Did the model drive a Mercedes?  N

# e1 14 b
The actress’ tailor, who the model’s photographer complemented, drove a Mercedes.
? Did the tailor complement the photographer?  N

# e1 14 c
The model, whose photographer complemented the actress’ tailor, drove a Mercedes.
? Did the model drive a Mercedes?  Y

# e1 14 d
The actress, whose tailor the model’s photographer complemented, drove a Mercedes.
? Did the designer complement the photographer?  N

# e1 15 a
The executive’s employee, who tripped the manager’s friend, bought a new car.
? Did the employee trip the friend?  Y

# e1 15 b
The manager’s friend, who the executive’s employee tripped, bought a new car.
? Did the friend buy a new car?  Y

# e1 15 c
The executive, whose employee tripped the manager’s friend, bought a new car.
? Did the employee trip the friend?  Y

# e1 15 d
The manager, whose friend the executive’s employee tripped, bought a new car.
? Did the manager buy a new car?  Y

# e1 16 a
The parent’s babysitter, who contacted the clown’s therapist, read the book.
? Did the babysitter read the book?  Y

# e1 16 b
The clown’s therapist, who the parent’s babysitter contacted, read the book.
? Did the therapist contact the babysitter?  N

# e1 16 c
The parent, whose babysitter contacted the clown’s therapist, read the book.
? Did the therapist read the book?  N

# e1 16 d
The clown, whose therapist the parent’s babysitter contacted, read the book.
? Did the babysitter contact the clown?  N

# e1 17 a
The king’s speechwriter, who impressed the general’s brother, saw the fire.
? Did the king impress the general?  N

# e1 17 b
The general’s brother, who the king’s speechwriter impressed, saw the fire.
? Did the general see the fire?  N

# e1 17 c
The king, whose speechwriter impressed the general’s brother, saw the fire.
? Did the king impress the brother?  N

# e1 17 d
The general, whose brother the king’s speechwriter impressed, saw the fire.
? Did the brother see the fire?  N

# e1 18 a
The soldier’s commander, who shot the minister’s son, spoke five languages.
? Did the minister speak five languages?  N

# e1 18 b
The minister’s son, who the soldier’s commander shot, spoke five languages.
? Did the commander shoot the son?  Y

# e1 18 c
The soldier, whose commander shot the minister’s son, spoke five languages.
? Did the soldier speak five languages?  Y

# e1 18 d
The minister, whose son the soldier’s commander shot, spoke five languages.
? Did the commander shoot the son?  Y

# e1 19 a
The girl’s mother, who met the victim’s murderer, visited Australia last year.
? Did the mother meet the murderer?  Y

# e1 19 b
The victim’s murderer, who the girl’s mother met, visited Australia last year.
? Did the murderer visit Australia last year?  Y

# e1 19 c
The girl, whose mother met the victim’s murderer, visited Australia last year.
? Did the mother meet the murderer?  Y

# e1 19 d
The victim, whose murderer the girl’s mother met, visited Australia last year.
? Did the girl visit Australia last year?  N

# e1 20 a
The journalist’s editor, who criticized the director’s cameraman, had many enemies.
? Did the editor have many enemies?  Y

# e1 20 b
The director’s cameraman, who the journalist’s editor criticized, had many enemies.
? Did the editor criticize the cameraman?  Y

# e1 20 c
The journalist, whose editor criticized the director’s cameraman, had many enemies.
? Did the journalist have many enemies?  Y

# e1 20 d
The director, whose cameraman the journalist’s editor criticized, had many enemies.
? Did the editor criticize the cameraman?  Y

# e1 21 a
The actor’s understudy, who telephoned the singer’s agent, lived in Hawaii.
? Did the actor telephone the agent?  N

# e1 21 b
The singer’s agent, who the actor’s understudy telephoned, lived in Hawaii.
? Did the understudy live in Hawaii?  N

# e1 21 c
The actor, whose understudy telephoned the singer’s agent, lived in Hawaii.
? Did the understudy live in Hawaii?  N

# e1 21 d
The singer, whose agent the actor’s understudy telephoned, lived in Hawaii.
? Did the singer live in Hawaii?  Y

# e1 22 a
The accountant’s consultant, who hired the banker’s programmer, broke the computer.
? Did the accountant break the computer?  N

# e1 22 b
The banker’s programmer, who the accountant’s consultant hired, broke the computer.
? Did the banker hire the consultant?  N

# e1 22 c
The accountant, whose consultant hired the banker’s programmer, broke the computer.
? Did the consultant hire the programmer?  Y

# e1 22 d
The banker, whose programmer the accountant’s consultant hired, broke the computer.
? Did the accountant hire the banker?  N

# e1 23 a
The saleslady’s supervisor, who deceived the owner’s neighbor, just got married.
? Did the supervisor deceive the neighbor?  Y

# e1 23 b
The owner’s neighbor, who the saleslady’s supervisor deceived, just got married.
? Did the supervisor deceive the neighbor?  Y

# e1 23 c
The saleslady, whose supervisor deceived the owner’s neighbor, just got married.
? Did the owner just get married?  N

# e1 23 d
The owner, whose neighbor the saleslady’s supervisor deceived, just got married.
? Did the saleslady just get married?  N

# e1 24 a
The chauffeur’s passenger, who hated the boss’ stockbroker, heard the news.
? Did the chauffe hear the news?  N

# e1 24 b
The boss’ stockbroker, who the chauffeur’s passenger hated, heard the news.
? Did the passenger hear the news?  N

# e1 24 c
The chauffeur, whose passenger hated the boss’ stockbroker, heard the news.
? Did the passenger hate the boss?  N

# e1 24 d
The boss, whose stockbroker the chauffeur’s passenger hated, heard the news.
? Did the passenger hate the stockbroker?  Y
Appendix 2: experiment 2 stimuli

# e2 1 a
The secretary who sent the student to the administrator talked to the librarian.
? Did the secretary talk to the librarian?  Y

# e2 1 b
The student who the secretary sent to the administrator talked to the librarian.
? Did the secretary send the administrator?  N

# e2 1 c
The administrator to whom the secretary sent the student talked to the librarian.
? Did the administrator send someone to the student?  N

# e2 2 a
The spy who revealed the mole to the defector confessed under pressure.
? Did the spy reveal the mole?  Y

# e2 2 b
The mole who the spy revealed to the defector confessed under pressure.
? Did the spy reveal someone to the defector?  Y

# e2 2 c
The defector to whom the spy revealed the mole confessed under pressure.
? Did the spy confess under pressure?  N

# e2 3 a
The guard who presented the prisoner to the judge remained emotionless.
? Did the prisoner present someone to the judge?  N

# e2 3 b
The prisoner who the guard presented to the judge remained emotionless.
? Did the prisoner remain emotionless?  Y

# e2 3 c
The judge to whom the guard presented the prisoner remained emotionless.
? Did the judge present the prisoner?  N

# e2 4 a
The acrobat who passed the ballerina to the clown smiled at the audience.
? Did the acrobat smile at the audience?  Y

# e2 4 b
The ballerina who the acrobat passed to the clown smiled at the audience.
? Did the acrobat pass the clown?  N
The clown to whom the acrobat passed the ballerina smiled at the audience.
Did the acrobat pass the ballerina to someone? Y

The agent who introduced the singer to the guitarist knew everyone in town.
Did the agent introduce the singer? Y

The singer who the agent introduced to the guitarist knew everyone in town.
Did the singer introduce someone to the guitarist? N

The guitarist to whom the agent introduced the singer knew everyone in town.
Did the agent know everyone in town? N

The elf who guided the adventurer to the wizard did not know any magic.
Did the elf guide someone to the wizard? Y

The adventurer who the elf guided to the wizard did not know any magic.
Did the adventurer know any magic? N

The wizard to whom the elf guided the adventurer did not know any magic.
Did the elf guide the adventurer? Y

The doctor who referred the patient to the specialist feared terrible sickness.
Did the doctor refer the patient? Y

The patient who the doctor referred to the specialist feared terrible sickness.
Did the doctor refer someone to the specialist? Y

The specialist to whom the doctor referred the patient feared terrible sickness.
Did the patient fear terrible sickness? N

The kidnapper who returned the child to the mother felt bad about the
whole thing.
? Did the kidnapper return someone to the mother?  Y

# e2 8 b
The child who the kidnapper returned to the mother felt bad about the whole thing.
? Did the kidnapper feel bad about the whole thing?  N

# e2 8 c
The mother to whom the kidnapper returned the child felt bad about the whole thing.
? Did the mother return the child?  N

# e2 9 a
The crier who summoned the knight to the king suspected an emergency.
? Did the knight suspect an emergency?  N

# e2 9 b
The knight who the crier summoned to the king suspected an emergency.
? Did the crier summon the knight?  Y

# e2 9 c
The king to whom the crier summoned the knight suspected an emergency.
? Did the king summon someone to the knight?  N

# e2 10 a
The soldier who surrendered the captive to the officer wore a ragged uniform.
? Did the soldier surrender the captive?  Y

# e2 10 b
The captive who the soldier surrendered to the officer wore a ragged uniform.
? Did the captive surrender someone to the officer?  N

# e2 10 c
The officer to whom the soldier surrendered the captive wore a ragged uniform.
? Did the officer wear a ragged uniform?  Y

# e2 11 a
The gangster who provided the operative to the kingpin asked for thirty thousand.
? Did the operative provide someone for the kingpin?  N

# e2 11 b
The operative who the gangster provided to the kingpin asked for thirty thousand.
? Did the operative ask for thirty thousand?  Y
# e2 11 c
The kingpin to whom the gangster provided the operative asked for thirty thousand.
? Did the gangster provide the kingpin?  N

# e2 12 a
The father who entrusted the girl to the teacher believed in education.
? Did the father believe in education?  Y

# e2 12 b
The girl who the father entrusted to the teacher believed in education.
? Did the girl entrust the father?  N

# e2 12 c
The teacher to whom the father entrusted the girl believed in education.
? Did the father entrust someone to the teacher?  Y

# e2 13 a
The recruiter who forwarded the applicant to the boss liked the business.
? Did the recruiter forward the boss?  N

# e2 13 b
The applicant who the recruiter forwarded to the boss liked the business.
? Did the recruiter forward someone to the boss?  Y

# e2 13 c
The boss to whom the recruiter forwarded the applicant liked the business.
? Did the applicant like the business?  N

# e2 14 a
The theorist who related the historian to the economist achieved posthumous fame.
? Did the theorist relate someone to the economist?  Y

# e2 14 b
The historian who the theorist related to the economist achieved posthumous fame.
? Did the economist achieve posthumous fame?  N

# e2 14 c
The economist to whom the theorist related the historian achieved posthumous fame.
? Did the theorist relate the historian?  Y

# e2 15 a
The broker who connected the buyer to the seller was well-regarded on Wall Street.
? Was the buyer well-regarded on Wall Street?  N

# e2 15 b
The buyer who the broker connected to the seller was well-regarded on Wall Street.
? Did the broker connect the buyer?  Y

# e2 15 c
The seller to whom the broker connected the buyer was well-regarded on Wall Street.
? Did the seller connect someone to the buyer?  N

# e2 16 a
The host who suggested the comedian to the producer had a good nose for material.
? Did the host suggest the comedian?  Y

# e2 16 b
The comedian who the host suggested to the producer had a good nose for material.
? Did the comedian suggest someone to the producer?  N

# e2 16 c
The producer to whom the host suggested the comedian had a good nose for material.
? Did the producer have a good nose for material?  Y

# e2 17 a
The caterer who supplied the waiter to the organizer chatted with the guests.
? Did the caterer supply someone to the waiter?  N

# e2 17 b
The waiter who the caterer supplied to the organizer chatted with the guests.
? Did the waiter chat with the guests?  Y

# e2 17 c
The organizer to whom the caterer supplied the waiter chatted with the guests.
? Did the organizer supply the caterer?  N

# e2 18 a
The policeman who directed the driver to the mechanic was familiar with the area.
? Was the policeman familiar with the area?  Y

# e2 18 b
The driver who the policeman directed to the mechanic was familiar with the area.
? Did the policeman direct the mechanic?  N

# e2 18 c
The mechanic to whom the policeman directed the driver was familiar with the area.

? Did the policeman direct someone to the mechanic?  Y
Appendix 3: experiment 3 stimuli

# e3 1 a
The officer who pacified the captor for the hostage held a knife.
? Did the officer hold a knife?  Y

# e3 1 b
The captor who the officer pacified for the hostage held a knife.
? Did the officer pacify someone for the hostage?  Y

# e3 1 c
The hostage for whom the officer pacified the captor held a knife.
? Did the officer pacify the hostage?  N

# e3 2 a
The accountant who hired the assistant for the boss wore a grey suit.
? Did the boss hire the assistant  N

# e3 2 b
The assistant who the accountant hired for the boss wore a grey suit.
? Did the assistant wear a grey suit?  Y

# e3 2 c
The boss for whom the accountant hired the assistant wore a grey suit.
? Did the boss wear a grey suit?  Y

# e3 3 a
The secretary who contacted the patient for the doctor woke up early.
? Did the patient wake up early?  N

# e3 3 b
The patient who the secretary contacted for the doctor woke up early.
? Did the patient contact the doctor?  N

# e3 3 c
The doctor for whom the secretary contacted the patient woke up early.
? Did the secretary contact someone for the doctor?  Y

# e3 4 a
The cameraman who videotaped the actor for the director fell down the stairs.
? Did the cameraman videotape someone for the director?  Y

# e3 4 b
The actor who the cameraman videotaped for the director fell down the stairs.
? Did the actor fall down the stairs?  Y

# e3 4 c
The director for whom the cameraman videotaped the actor fell down the
stairs.
? Did the camerman fall down the stairs?  N

# e3 5 a
The babysitter who watched the girl for the parent saw the movie.
? Did the babysitter see the movie?  Y

# e3 5 b
The girl who the babysitter watched for the parents saw the movie.
? Did the girl watch someone for the parents?  N

# e3 5 c
The parents for whom the babysitter watched the girl saw the movie.
? Did the babysitter parents watch the girl?  N

# e3 6 a
The detective who questioned the suspect for the chief closed the door.
? Did the chief question the suspect?  N

# e3 6 b
The suspect who the detective questioned for the chief closed the door.
? Did the detective close the door?  N

# e3 6 c
The chief for whom the detective questioned the suspect closed the door.
? Did the chief close the door?  Y

# e3 7 a
The employee who trained the cashier for the supervisor stole from the store.
? Did the cashier steal from the store?  N

# e3 7 b
The cashier who the employee trained for the supervisor stole from the store.
? Did the employee train the cashier?  Y

# e3 7 c
The supervisor for whom the employee trained the cashier stole from the store.
? Did the cashier train someone for the manager?  N

# e3 8 a
The intern who observed the subject for the researcher left work early.
? Did the subject observe someone for the researcher?  N

# e3 8 b
The subject who the intern observed for the researcher left work early.
? Did the intern leave work early?  N
# e3 8 c
The researcher for whom the intern observed the subject left work early. ? Did the researcher leave work early? Y

# e3 9 a
The agent who called the lawyer for the celebrity drove a Porsche. ? Did the agent drive a Porsche? Y

# e3 9 b
The lawyer who the agent called for the celebrity drove a Porsche. ? Did the agent call someone for the celebrity? Y

# e3 9 c
The celebrity for whom the agent called the lawyer drove a Porsche. ? Did the celebrity call the lawyer? N

# e3 10 a
The jock who punched the bully for the nerd failed history last year. ? Did the jock punch the bully? Y

# e3 10 b
The bully who the jock punched for the nerd failed history last year. ? Did the bully fail history last year? Y

# e3 10 c
The nerd for whom the jock punched the bully failed history last year. ? Did the jock fail history last year? N

# e3 11 a
The messenger who summoned the soldier for the general tripped on a rock. ? Did the soldier trip on a rock? N

# e3 11 b
The soldier who the messenger summoned for the general tripped on a rock. ? Did the messenger summon the general? N

# e3 11 c
The general for whom the messenger summoned the soldier tripped on a rock. ? Did the messenger summon someone for the general? Y

# e3 12 a
The worker who interviewed the applicant for the executive broke the elevator. ? Did the worker break the elevator? Y

# e3 12 b
The applicant who the worker interviewed for the executive broke the
elevator.
? Did the worker interview someone for the executive?  Y

# e3 12 c
The executive for whom the worker interviewed the applicant broke the elevator.
? Did the worker interview the applicant?  Y

# e3 13 a
The owner who identified the robber for the policeman lived in Florida.
? Did the policeman identify the robber?  N

# e3 13 b
The robber who the owner identified for the policeman lived in Florida.
? Did the policeman live in Florida?  N

# e3 13 c
The policeman for whom the owner identified the robber lived in Florida.
? Did the owner live in Florida?  N

# e3 14 a
The counselor who advocated the tutor for the student rang the doorbell.
? Did the counselor ring the doorbell?  Y

# e3 14 b
The tutor who the counselor advocated for the student rang the doorbell.
? Did the tutor advocate the counselor?  N

# e3 14 c
The student for whom the counselor advocated the tutor rang the doorbell.
? Did the counselor advocate someone for the student?  Y

# e3 15 a
The guard who watched the traitor for the king went down to the basement.
? Did the traitor watch someone for the king?  N

# e3 15 b
The traitor who the guard watched for the king went down to the basement.
? Did the traitor go down to the basement?  Y

# e3 15 c
The king for whom the guard watched the traitor went down to the basement.
? Did the soldier go down to the basement?  N

# e3 16 a
The expert who helped the beginner for the instructor forgot to clean
up.
? Did the expert help the beginner?  Y

# e3 16 b
The beginner who the expert helped for the instructor forgot to clean up.
? Did the expert forget to clean up?  N

# e3 16 c
The instructor for whom the expert helped the beginner forgot to clean up.
? Did the instructor forget to clean up?  N

# e3 17 a
The nurse who recommended the dermatologist for the patient looked at the file.
? Did the nurse look at the file?  Y

# e3 17 b
The dermatologist who the nurse recommended for the patient looked at the file.
? Did the nurse look at the file?  N

# e3 17 c
The patient for whom the nurse recommended the dermatologist looked at the file.
? Did the nurse recommend someone for the patient?  Y

# e3 18 a
The waiter who brought the manager for the customer spilled the drinks.
? Did the manager bring someone for the customer?  N

# e3 18 b
The manager who the waiter brought for the customer spilled the drinks.
? Did the manager spill the drinks?  Y

# e3 18 c
The customer for whom the waiter brought the manager spilled the drinks.
? Did the customer spill the drinks?  Y
Chapter 6

Kinds of probabilistic sentence processing theories

The proposals of chapters 2 and 3 seek to connect admittedly abstract ideas with specific (psycho-) linguistic issues. Oftentimes the distance between these two is very great. An idea like “cognition is really information processing” might seem rather far removed from a question like “why do people spend more time reading embedded verbs in object-extracted than in subject-extracted relative clauses” – so far removed that defending the existence of even one connection between them appears to be impossible. Nonetheless, that is what chapters 2 and 3 attempt to do. They attempt to make plausible a single connection between the abstract view of cognition as information-processing, itself subject to information theory, and the empirical concerns of behavioral and linguistic inquiry.

The present chapter considers other routes between these two kinds of ideas. It compares the ideas presented in earlier chapters with alternative proposals, charting a rough map of the space of related probabilistic sentence processing theories. This map is three dimensional. The first dimension, Empirical coverage (subsection 6.1) is obviously important, but the discussion here will add little. The second, Quantity proposed as a psycholinguistic model (subsection 6.2) categorizes the mathematical objects used in probabilistic sentence processing theories. This section contributes a clarification of the differences and deep similarities between the ideas proposed in this dissertation and a few closely-related ones. The third dimension, Grammar type (section 6.3) speaks to the interplay between processor and grammar in these theories. It emphasizes the difference between syntactic ambiguities that make a perceptible difference in semantic interpretation and those that do not.
6.1 Empirical coverage

Theories are accounts of real world phenomena, of course, and the most important comparison is their coverage of those phenomena. Unfortunately, the empirical foci of existing probabilistic processing theories diverge radically from one another. On this point, relatively little can be said that has not been said before: the theoretical psycholinguistics community should standardize on a set of well-established phenomena. One systematic and fairly theory-neural candidate is the set of ambiguity-resolution phenomena cataloged in Lewis (1993). This collection encompasses the prior work of Gibson (1991) and Cowper (1976).

In chapter 3 the Keenan-Conrie (1977) Accessibility Hierarchy (AH) provides an attractively organized empirical domain against which the entropy reduction hypothesis 1 is tested. The failure to find oblique-extracted relative clause harder than indirect object-extracted ones in chapter 5 casts doubt on the AH. It may be that the relative clause structures listed on the AH are not naturally related in an accurate processing theory. This empirical finding revises the initial hypothesis, motivated by Strong Competence (cf. section 3.1.1) that they are.

6.2 Quantity proposed as a psycholinguistic model

Sentence processing theories differ on which mathematical object they use to model the empirical phenomena of human sentence understanding. One class of theories uses ranking.

6.2.1 Ranking theories

Iconified by the work of Jurafsky and Narayanan (Jurafsky, 1996; Narayanan and Jurafsky, 1998; Narayanan and Jurafsky, 2001), but also subsuming the proposals of Gibson (1991) and Stevenson and Smolensky (2003), ranking theories propose that a grammar creates a kind of "n-best list" of (partial) analyses of an incomplete sentence prefix. The scores on this ranked list can come either from frequencies encoded as probabilities in a probabilistic grammar or from a linguistic notion of markedness that is related to frequency only indirectly. In Gibson's (1991) ranked parallel model analyses are ranked by weighted violations of grammatically-motivated constraints. A ranking theory distinguishes two cases: the case where a new word is added to the prefix string under consideration and the ranking stays the same, and the case where a new word is added and the ranking changes. Typically the prediction is that if the ranking changes\(^1\), human sentence processing difficulty

\(^1\)In the presence of underspecified syntactic analyses the question of whether a particular analysis is the same or not between successful words becomes interesting. Often a ranked tree \(t_2\) is thought to be 'the same'
is predicted. Ranking theories can be refined by proposing that the degree of difficulty is proportional to some measure of how much the ranking changed from word to word.

The ranking perspective also accommodates Vasishth and Kruijff (2001) where a sequence of partial analyses is stored, and there is a fixed processing cost for indexing farther back into this sequence. In this model, the ranks are indices of the sequence and the ranking factors are unrelated to frequency.

Indeed, the category of ranking theories is so broad as to even include serial parsing theories (Frazier and Fodor, 1978). Roark’s (1999) probabilistic top-down parser demonstrates particularly cleanly how ranking theories generalize traditional serial parsing models by increasing the width of the ranking \( n \) beyond 1. The case where \( n = 1 \) is pursued in Chater, Crocker, and Pickering (1998) et seq.

6.2.2 Other scalar functions of a probabilistic grammar and a prefix string

Prefix probability

Rather than using a probabilistic grammar to rank an \( n \)-best list and then deriving predictions from changes in that object, Hale (2001) predicts sentence processing difficulty by computing a single value directly from a probabilistic grammar and a prefix string. This value is the surpsiral of the next word. This proposal makes use of Stolcke’s (1995) probabilistic Earley parser to calculate the prefix probability \( \alpha \) of an initial string \( w_0 \ldots w_i \) generated by a probabilistic grammar \( G \).

\[
\alpha_i = \sum_{d \text{ is a derivation on } G} P(d) \quad (6.1)
\]

and the string \( w_0 \ldots w_i \) is a left-prefix of \( d \)'s yield

Hale (2001) defines cognitive load as the surprisal of the event that a particular possible word follows a particular prefix, namely \( \log \left( \frac{\alpha_{n-1}}{\alpha_n} \right) \). This definition is motivated by the intuition that more time is spent reading when our expectations are not confirmed. Table 6.1 records the essence of the proposal.

The prefix probability goes down when smaller and smaller subsets of rules become necessary to analyse the given prefix on the given grammar. One can say a rule has been categorically as a previous tree \( t_1 \) if \( t_1 \) is a subtree of \( t_2 \). More graded notions of tree-sameness can be given (Courcelle, 1983; Kuroda, 1987) although the difference between elaboration and reanalysis is probably best treated at a semantic level.
Hale (2001) $\log\left(\frac{a^n-1}{a^n}\right)$

Table 6.1: Table of probabilistic processing theories (to be expanded)

“added” to a parser state when all viable derivations use that rule. At this point, the probability of all derivations contributing to the prefix probability are affected by the added rule’s probability. For this reason, in unambiguous sentences, the amount of predicted reading effort is proportional to the logarithm of the probability of *new rules* added upon hearing a word. Where many rules are simultaneously added – for instance, at the ends of large constituents – much difficulty is predicted. Under grammatical assumptions from GPSG, this leads to misplaced predictions in object-extracted relative clauses like the one shown in figure 6.1.

Figure 6.1: Object relative clause

In the course of Earley-parsing the syntactic analysis depicted in (6.1), only one new rule is added at the embedded verb “sent”. The rule is

$$VP/NP \rightarrow V3 \ NP/NP \ PP$$  \hspace{1cm} (6.2)

which might be glossed as “a verb phrase missing an object noun phrase contains a verb of subcategorization class three, a phonetically null trace, and a prepositional phrase complement.” By contrast, at the end of the subject noun phrase a slew of new rules can be completely recognized, quite apart from any prediction. An unwelcome consequence is that a large amount of load is predicted at “editor” while little is predicted at “sent”, where it actually occurs for human readers. Under a binary phrase structure regime, this predicts reading time at roughly the same places a thematic dependency account (Gibson, 2000)
would\textsuperscript{2}. However, where the correlation between completed constituents and resolved dependencies breaks down, the predictions diverge. This is exactly what happens in the rule 6.2 where, intuitively the verb “sent” has two dependents, the displaced NP “the reporter” and the complement PP “to the editor”. The virtue of the dependency account, where the width of the sent–reporter dependency plays a key role, cannot be duplicated by an account based on completed constituents – at least not without radically altering the constituency claims of figure 6.1. Note also that the opposite approach, predicting reading time at “proposal time” when a rule is first motivated by top-down considerations, runs into an analogous difficulty.

So, while simple and efficiently-computable the prefix probability theory is thwarted by at least one empirical difficulty.

**KL distance**

Another natural probabilistic processing hypothesis is that human processing difficulty is proportional to the Kullback-Leibler (KL) distance between the distribution on trees compatible with a n-word prefix string and the distribution on trees compatible with the n + 1-word prefix string. However, Roger Levy (p.c.) has shown that this theory is actually equivalent to the prefix probability theory. Define as \( T_n \) the set of all complete trees whose yield includes a given prefix string of length \( n \). Denote by \( p_n \) the probability of a parse tree given a prefix of length \( n \) (i.e. a distribution on surviving parses after \( n \) words have been heard). Then, since the underlying grammar is the same for the two prefix-constrained probability spaces, for \( t_1, t_2 \in T_{n+1} \) we have

\[
\frac{p_n(t_1)}{p_n(t_2)} = \frac{p_{n+1}(t_1)}{p_{n+1}(t_2)}.
\]

That is to say, the probability ratio is unaffected by hearing the next word as long as neither tree is eliminated. It is also true by definition that

\[
1 = \sum_{t \in T_{n+1}} p_{n+1}(t).
\]

Define \( Q_{n+1} \) to be the probability mass (for the \( p_n \) distribution) of trees eliminated by the \( n + 1 \)th member of the prefix string. That is,

\[
Q_{n+1} = \text{def} \sum_{t \in T_n, \ t \not\in T_{n+1}} p_n(t)
\]

since no new trees are ever added, all the surviving trees are in the complement set:

\textsuperscript{2}The elaborations of context-free grammar required for a dependency interpretation are discussed by Miller (1999).
\[ 1 - Q_{n+1} = \text{def} \sum_{t \in T_{n+1}} p_n(t). \] (6.6)

From facts 6.3 and 6.6 it follows that the ratio \( \frac{p_n(t)}{p_{n+1}(t)} \) must equal \( 1 - Q_{n+1} \) for all \( t \in T_{n+1} \).

So the KL distance is

\[
D(p_{n+1} \| p_n) = \sum_{t \in T_{n+1}} p_{n+1} \log \left( \frac{1}{1 - Q_{n+1}} \right)
= -\log(1 - Q_{n+1})
\]

This value, \( -\log(1 - Q_{n+1}) \), is the same as the ratio of prefix probabilities \( \alpha_n \) and \( \alpha_{n+1} \) in Hale (2001). So in fact, these two theories are exactly the same. Table 6.2 is updated to reflect this result.

<table>
<thead>
<tr>
<th>Hale (2001)</th>
<th>( \log \left( \frac{\alpha_{n+1}}{\alpha_n} \right) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>KL-distance (Levy)</td>
<td>( D(p_{n+1} | p_n) = \log \left( \frac{\alpha_{n+1}}{\alpha_n} \right) )</td>
</tr>
</tbody>
</table>

Table 6.2: Table of probabilistic processing theories (to be expanded)

**Conditional entropy reduction and Mutual Information**

The proposals of chapters 2 and 3 are unlike KL-distance. They suggest that processing difficulty is proportional to the change in average uncertainty about the whole sentence, based on the particular word heard. That is, the reduction in the entropy of the compatible derivations given knowledge of a related random variable corresponding to the next word. If \( X \) is the entropy of the derivations compatible with a prefix of length \( n \) and \( Y \) is the word in position \( n \) the general form of the entropy reduction hypothesis 1 is

\[ \max(0, H(X) - H(X|Y = y)) \] (6.7)

The quantity \( H(X) - H(X|Y = y) \) is not guaranteed to be positive because sometimes the outcome \( Y = y \) takes the parser into a region of derivation-space that is more uncertain than on average. The theory avoids predicting negative processing effort by ignoring negative predictions (hence the use of \( \max \)). One would naturally suppose that all words convey information. The difference between 6.7 and the mutual information between the derivation and the \( n^{th} \) position of the prefix string 6.8 – which is necessarily positive – is that the latter averages over the prefix probabilities \( p_y \) whereas the former does not.
\[ H(X) - H(X|Y) = I(X; Y) \] (6.8)

This theory has its own disadvantages. It is insensitive to the particular word that is actually perceived at position \( n \), referring only to the average information conveyed by whatever occupies that position. \( I(X; Y) \) could be evaluated by calculating \( H(X) \) and then \( H(X|Y = y) \) for all \( y \), but this latter term would need to be weighted by \( p_y \) for all \( Y = y \). For many grammar formalisms, including Minimalist Grammars, efficiently computing the prefix probabilities \( p_y \) is an open problem.\(^3\) Having distinguished 6.8 from the other proposals, the table 6.3 can be filled out.

### Table 6.3: Table of probabilistic processing theories (final)

<table>
<thead>
<tr>
<th>Theory</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hale (2001)</td>
<td>[ \log \left( \frac{a_{n+1}}{a_n} \right) ]</td>
</tr>
<tr>
<td>KL-distance (Levy)</td>
<td>[ D(p_{n+1}</td>
</tr>
<tr>
<td>entropy reduction (chapter 3)</td>
<td>( H(X) - H(X</td>
</tr>
<tr>
<td>mutual information</td>
<td>( H(X) - H(X</td>
</tr>
</tbody>
</table>

### 6.3 Grammar type

A crucial ingredient of any probabilistic sentence processing theory is the grammar the comprehender uses to analyze input sentences. For this task, chapter 2 assumed various particular context-free grammars. Chapter 3 generalized the proposal by adopting a mildly context-sensitive grammar. However, neither chapter offered any sort of semantic interpretation component. While this could be straightforwardly added to either (perhaps by associating compositional rules building lambda-expressions at each derivation tree node) it was asserted earlier in chapter 2 that

It suffices to consider only syntactic processing, since semantic rules are taken to be in one-to-one correspondence with syntactic rules (Steedman, 2000).

This claim was meant as a simplifying assumption. For it to be completely upheld in a realistic grammar of any significant coverage would be surprising. For instance, Charniak, Johnson, and Caraballo (1998) found that sentences of length 40 or less required on average 1.2 million chart entries per sentence, on the implicit Penn Treebank grammar. Keeping

\(^3\) An efficient algorithm for prefix probabilities has been found for TAG (Nederhof, Sarkar, and Satta, 1998).
in mind that the chart representation is an optimally compact representation of derivational ambiguities, it becomes highly implausible that all of these ambiguities correspond to differences of interpretation. Rather, some of these ambiguities must be purely syntactic in nature, corresponding to no psychologically- perceptible difference in interpretation. In combination with the one-to-one syntax/semantics correspondence proposed on page 13, the entropy reduction hypothesis 1 is blindly sensitive to all such syntactic ambiguities.

Perhaps the simplest possible case of this problem (pointed out to the author by Chris Manning) is that of proper names. A realistic grammar including a category for proper names is likely to have a large number of alternative rewrites as suggested in figure 6.2.

ProperName → Bill | Ted | Paul | Scott | Ed | Géraldine | Katherine | Julia | …

Figure 6.2: A grammar rule for proper names

Clearly a rule like this is necessary to achieve coverage of text samples that mention a wide variety of people. Equally clearly, though, the distribution on alternatives for the rule in figure 6.2 has very little linguistic relevance—people talk about other people according to their real-world needs and desires, unconstrained by any grammatical restriction. This distribution might as well be equiprobable—the maximum entropy distribution. So, upon hearing a discourse like, “You’ll never guess who I saw at the concert last night. It was ___” the entropy reduction hypothesis predicts maximal comprehension effort resolving the substructure of the derivation rooted in ProperName. This is a spurious prediction since people have no trouble understanding the continuation “...it was Julia.”

The lesson of this thought-experiment is that some ambiguities present in syntactic grammars should not matter for processing. But which ones do? Chomsky’s (1956) proposal was that only the ambiguities that correspond to differences in interpretation should matter. In discussing two derivations of “they are flying planes” he remarks:

When the simplest grammar automatically provides nonequivalent derivations for some sentence, we say that we have a case of constructional homonymity and we can suggest this formal property as an explanation for the semantic ambiguity in question....

One way to test the adequacy of a grammar is by determining whether or not the cases of constructional homonymity are actually cases of semantic ambiguity...

(Chomsky, 1956, 118)

Results like those of Charniak, Johnson, and Caraballo (1998) suggest that wide-coverage grammars that are adequate in this respect are a long way off. In the meantime, it would be possible to evaluate Chomsky’s proposal by deciding, on a per-rule basis which ambiguities could potentially make a semantic difference and which cannot—this decision would be objective in the presence of an explicit formal semantics. The same entropy reduction
hypothesis 1 would then be applicable to this subset of derivational decisions. If this subset
excludes decisions such as the choice of right-hand side for the rule in figure 6.2 the spurious
prediction of high load following “It was...” can be avoided.

The implication for probabilistic sentence processing theories is that they can differ
on the subset of grammatical ambiguities that matter for processing. This is the ‘grain size’
problem noted by Desmet and Gibson (2003). As the grammar formalism varies, so too does
the exact definition of a single ambiguity. What might be a single choice in one grammar
could be distributed across several choices in another. For instance, in chapter 3 processing
predictions derived from the Kaynian analysis of relative clauses correlated better with
repetition accuracy scores than did predictions derived from the adjunction analysis. This
could only be because the two grammars explicitly specify a different set of ambiguities for
the same sentences.
Chapter 7

Conclusions and Future Work

7.1 Conclusions

This dissertation has proposed a definition of computational work performed in the course of incrementally parsing sentences generated by probabilistic grammars. It explored the consequences of this definition as a cognitive model of human sentence processing.

Chapter 2 found that this definition rationalizes a variety of well-studied sentence processing phenomena, like garden-pathing, the subject/object-extraction asymmetry and the NP/S vs. NP/Z ambiguity.

Chapter 3 extended this proposal beyond just subject- and object-extraction to the entire Accessibility Hierarchy (Keenan and Comrie, 1977) of relativizable grammatical relations. Chapter 3 also integrated existing techniques from formal language theory to apply the definition of computational work to the more expressive Minimalist Grammars (Stabler, 1997) formalism. This made it possible to compare rival grammatical proposals about the structure of relative clauses, leading to the conclusion that Kayne's (1994) proposal correlates better with the processing evidence that does the traditional adjunction analysis.

Chapter 5 examined, in detail, the online reading comprehension difficulty of constructions on the Accessibility Hierarchy, confirming predictions about genitivity and (in)directness, but failing to confirm a prediction about obliqueness. Chapter 4 and appendix A collectively demonstrated that it is, in fact, possible to distill current proposals from the syntax literature down to a formal grammar that can be exhaustively parsed on real computers, and used to make psycholinguistic predictions.

Chapter 6 stepped back and put the proposals of earlier chapters into a three-dimensional perspective, along which probabilistic theories of sentence processing should be compared. They should certainly be compared along a dimension of empirical adequacy that is the same across theories. Their mathematical character—the particular value proposed as a cognitive model of sentence processing difficulty—should be compared. Finally, the
set of ambiguities defined by the grammars they employ should also be compared.

7.2 Future work

The cognitive model explored in this dissertation is only vaguely specified. In many ways this is its cardinal virtue – that there is no reliance on the usual apparatus of search heuristics, processing principles or auxiliary data structures to characterize the processor. On the view proposed in this dissertation, the processor can be anything that eventually decides on a set of grammatical analyses after each word. In future work, it would be interesting to see what new predictions follow from specific instantiations of such a processor. It could be that particular parsing algorithms (e.g. left-corner parsing (Rosenkrantz and Lewis, 1970; Resnik, 1992; Manning and Carpenter, 1997)) fit the view of disambiguation-work-as-entropy reduction better than others. Even more broadly, certain abstract computer architectures (e.g. connectionist networks (McClelland and Kawamoto, 1985; Steedman, 1999; Rohde, 2002)) may realize the proposed cognitive model better than others – and may require testable new auxiliary hypothesis to do so.

Beyond making the cognitive model more specific, another area that is left to future work is learning. This dissertation has offered no theoretical interpretation of the numerical parameters used in the probabilistic grammars needed to derive sentence processing predictions. It is simply the case that these numbers can represent frequencies and that they are also useful, in combination with the categorical part of probabilistic grammars, for deriving processing predictions. The questions that will most certainly need to figure in future work involve these probabilities. What is their relation to frequency? To linguistic markedness? Do they even need to have all the properties numbers have? If this information is part of our knowledge of language (as opposed to being a neatly separable part of the processor) it will be enlightening to determine how it is learned from the kind of input children actually receive, growing up.
Appendix A

Implementing a parser for Minimalist Grammars in a functional programming language

This appendix documents a parser for Minimalist Grammars (Stabler, 1997; Stabler and Keenan, 2000) used to obtain some of results presented in this thesis. Other such programs are listed at


The implementation described here is used mainly as a tool for calculating quantities involved in a hypothesis about human performance in sentence processing experiments. However, it can also be viewed as support – by example – for the claim that strongly-typed functional programming languages facilitate the development of natural language software to an even greater degree than do imperative or logic programming languages. This position is argued for by Ljunglöf (2002) and Skalka (1998) among others.

Overall design

The implemented program takes as input a Minimalist Grammar (MG) (see section A.1) and a string, and returns any derivations that generate the given string on the given grammar. To compute this function from input strings and grammars to sets of derivations, it uses the bottom-up algorithm described in chapter 4 of Harkema (2001)\(^1\).

Harkema’s bottom-up algorithm naturally generalizes the Cocke-Younger-Kasami algorithm (Younger, 1967) to MGs. Following Sikkel (1997) the distinctive part of the

\(^1\)The program extends Harkema’s recognizer in various ways by keeping track of derivations as well as implementing head-movement, affix-hopping and adjunction.
parsing algorithm is expressed as a deduction system over 'items' each symbolizing sets of trees that could be generated by a given MG (see section A.2). The effectiveness of this algorithm fundamentally follows from the observation (Michaelis, 1998) that MGs can be viewed as a kind of Multiple Context-Free Grammar (Seki et al., 1991).

To create an executable recognizer, the deduction rules of Harkema's algorithm need only to be combined with the memory and control structures of a deductive engine like the one presented by Shieber, Schabes, and Pereira (1995). Their agenda-driven chart parser (Kay, 1986) searches for proofs using deduction rules that constitute the specification of a particular parsing algorithm. In this kind of parser, a chart records previously-derived items so their existence is not repeatedly re-determined (see section A.3) while an agenda stores items whose consequences have not yet been explored (see section A.4).

The design of the parser follows Shieber, Schabes, and Pereira (1995) very closely, except that instead of Prolog, the language used is Objective Caml (see http://caml.inria.fr). Parser inferences from items to items are implemented by functions that return lists of solutions, instead of as nondeterministic predicates that can succeed several ways. Unification, which is a primitive operation in Prolog had to be implemented separately (see section A.5). Since Objective Caml can be compiled to machine code, however the resulting parser is still much faster than the Prolog version (see section A.6).

### A.1 Grammar

The parser accepts grammars written in exactly the same format as the Prolog version. This facilitates debugging and permits comparison between the two. This format allows the user to specify a set of lexical entries, a start category and additional axioms about acceptable kinds of adjunction. An LR parser to read these grammars was automatically generated using the Ocamllex and Ocamllyacc tools.

Discussion of a particular Minimalist Grammar for a fragment of English can be found in appendix 4.

### A.2 Items

In Minimalist Grammars, lexical entries and phrasal categories are made from the same basic ingredients; they involve lists of features. Features have names that are grammar-specific – for instance d for determiner, wh for WH-movement, or k for case features. This idea is given a natural expression in ML as an enumerated type.
\texttt{type feature =}
\begin{itemize}
  \item Select of name
  \item Category of name
  \item Attract of name
  \item Licensee of name
  \item Phonetic of name
  \item RIncorp of name
  \item LIncorp of name
  \item RAffhop of name
  \item LAffhop of name
\end{itemize}
\texttt{type flist = feature list}

Using this \texttt{type}, an item is simply a list of \textit{chains} where each chain is feature list annotated with information about spans of the input string.

\texttt{type index = Position of int}
\texttt{type yesno = No | Yes}
\texttt{type cinfo = \{sleft:index; srigh:index;}
\hspace{1cm}hleft:index; hright:index;
\hspace{1cm}cleft:index; cright:index;
\hspace{1cm}complexyesno;
\hspace{1cm}exposed: flist\}\}
\texttt{type chain = Chain of cinfo}
\texttt{type item = chain list}

The record fields whose names end in \texttt{-left} and \texttt{-right} refer to left and right boundaries of the specifier, head and complement respectively. The flag \texttt{complex} marks the difference between lexical and nonlexical categories. \texttt{exposed} is the subsequence of features still accessible to Harkema’s CKY deduction rules.

Since MG deduction rules “check off” features as they apply, the \texttt{exposed} features of chains only get smaller and smaller as a parse is constructed bottom-up. This means that the total set of all subsequences of features (and hence all \texttt{flist} values) can be determined statically, per-grammar and replaced by a single integer.

\subsection*{A.2.1 Variables}

Another complexity masked by the type definition given above is the issue of non-ground items. To save memory, a set of items can be abbreviated by leaving one or another
subpart uninstantiated. The meaning of such an item is the set of items that result from instantiating the non-ground item all possible ways. Prolog handles this implicitly. In ML, uninstantiated variables can be just another type.

\[
\text{type name} = \text{Const of string} \mid \text{Var of string}
\]

\[
\text{type feature} =
\begin{align*}
& \mid \text{Select of name} \\
& \mid \text{Category of name} \\
& \mid \text{Attract of name} \\
& \mid \text{License of name} \\
& \mid \text{Phonetic of name} \\
& \mid \text{RIncorp of name} \\
& \mid \text{LIncorp of name} \\
& \mid \text{RAffhop of name} \\
& \mid \text{LAffhop of name} \\
\end{align*}
\]

\[
\text{type flist} = \text{FListVar of string} \mid \text{FCons of feature} * \text{flist} \mid \text{FEmpty}
\]

\[
\text{type index} = \text{Position of int} \mid \text{PositionVar of string}
\]

\[
\text{type yesno} = \text{No} \mid \text{Yes} \mid \text{YesNoVar of string}
\]

\[
\text{type cinfo} = \{ \text{sleft: index}; \text{tright: index}; \text{cleft: index}; \text{cright: index}; \text{complex: yesno}; \text{exposed: flist} \}
\]

\[
\text{type chain} = \text{Chain of cinfo} \mid \text{ChainVar of string}
\]

\[
\text{type clist} = \text{CListVar of string} \mid \text{CCons of chain} * \text{clist} \mid \text{CEmpty}
\]

| CAppend of clist * clist

New types for uninstantiated lists are given above for \text{flist} and \text{clist}. These type are flexible enough to encode both items in the chart and agenda as well as the MG deduction rules themselves. For instance, the rule \text{r3} (the third case of merge) as written in the notation of Stabler and Køenan (2000) looks like this:

\[
\begin{align*}
\text{r3} : 
& \frac{s \cdot f \gamma, \alpha_1, \ldots, \alpha_k \quad t \cdot f \delta, \iota_1, \ldots, \iota_l}{s : \gamma; t : \delta, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}
\end{align*}
\]

and can be rendered in ML as:
let \( r3ant1 = CCons \) (Chain \{sleft=PositionVar "A0";sright=PositionVar "A";
   hleft=PositionVar "B0";hright=PositionVar "B";
   cleft=PositionVar "C0";cright=PositionVar "C";
   complex=YesNoVar "either";
   exposed=FCons (Select \{ Var "f" \},FListVar "gamma")\},
CListVar "alphas"));

let \( r3ant2 = CCons \) (Chain \{sleft=PositionVar "D0";sright=PositionVar "D1";
   hleft=PositionVar "D1";hright=PositionVar "D2";
   cleft=PositionVar "D2";cright=PositionVar "D";
   complex=YesNoVar "eitherprime";
   exposed=FCons (Category \{ Var "f" \}, FCons (FeatureVar "req", ListVar "delta")),CListVar "iotas"));

let \( r3consq = CCons \) (Chain \{sleft=PositionVar "A0";sright=PositionVar "A";
   hleft=PositionVar "B0";hright=PositionVar "B";
   cleft=PositionVar "C0";cright=PositionVar "C";
   complex=Yes;
   exposed=FListVar "gamma"\},
(CCons (Chain \{sleft=PositionVar "D0";sright=PositionVar "D";
   hleft=Position \((-1)\);hright=Position \((-1)\);
   cleft=Position \((-1)\);cright=Position \((-1)\);
   complex=Yes;
   exposed=FCons (FeatureVar "req",FListVar "delta")),
CAppend (CListVar "alphas",CListVar "iotas"))));

This typed representation is convenient for debugging. For instance, ML's strong type system protects the deduction-rule writer from confusing CLists with FLists.

At runtime however, all this rich typing information is thrown away. Each rule is converted to a single ‘term’ type so that a faster and simpler unification algorithm can be used (section A.5).

Indeed non-ground types are only actually used in items for string positions. For instance, the lexical entry for an empty category – which could appear anywhere – is inserted as a single item containing positions that are all variables. An alternative would be to pay a space penalty and insert all possible empty categories at every position \((i, i)\) for all input string positions \(i\). Likewise, all deduction rules whose consequent involves uninstantiated variables would then be interpreted as multiple solutions, paying another space penalty. Such an arrangement might make parsing inferences simpler at the cost of multiplying the amount of memory required for the chart and agenda.
A.3 The chart

The chart is the data structure where derived items are stored. Several implementation considerations enter into its design.

1. looking up items in the chart must be fast (subsection A.3.1)

2. derivations must somehow be recorded. (subsection A.3.2)

A.3.1 Lookup

In Minimalist Grammars, the applicability of structure-building rules is almost completely determined by the identity of the *first feature*. In terms of the type definitions of section A.2 this is the head of the exposed *f*list. Intuitively, if the first exposed feature of an item is a Select feature =d then the parser should attempt to apply a merge rule. To do that it needs to search the chart for entries whose exposed feature is the corresponding Category feature d. Likewise with movement features e.g. +wh will prompt a search for -wh.

To achieve this goal, the chart is indexed by first features. More specifically, the chart is a tuple including

- an array of lists of chart entries

  Each cell of the array holds a reversed list of entries whose items all share the same first-feature. Retrieving “all items with first feature +f” simply means returning one of these cells. Adding a new entry means re-setting an array cell to a new list whose head is the new entry and whose tail is the old contents of the array cell.

- a hash table mapping from first features to array indices

  Taking advantage of the functional nature of ML, this hash table is passed as a unapplied function.

Entries are never removed from the chart, which grows as the parser runs. Tighter indexing might be possible; for instance in the Earley parser of Stolcke (1995) the category of the “next” symbol and the rightmost bound of the previous symbol are both hashed. In CKY parsing, items are traditionally indexed using a two-dimensional array to exactly match (just) string positions.

However, since it is ultimately the deduction rules of a particular parsing algorithm that determine the kinds of queries that need to be performed on the chart, optimizing for one particular set of rules may be premature. The Minimalist Grammars formalism is still developing and, although inspired by the merge and move rules of Chomsky (1995a), many different structure-building rules are nonetheless being considered (Stabler, 2002). A
more sophisticated analysis that goes beyond the rule of thumb that compatibility along first exposed features winnows the field considerably would index on all possible properties, including string positions and the existence of licensee features buried in the chain list. Section A.5 presents experimental observations regarding the gains to be had in this direction.

A.3.2 Derivations

Perhaps the most natural way to encode derivations is to add an additional set-valued field to chart entries, and whenever a new derivation is found, assign to this field the union of the old derivation set with the new set. This is essentially the approach taken in the implemented parser, although, as Shieber, Schabes, and Pereira (1995) point out, such an approach forsakes all hope of polynomial complexity. If there are an exponential number of derivations, on such a naive approach, there will need to be an exponential amount of resetting these derivation fields to larger and larger sets.

Backpointers avoid this problem. Instead of storing a whole derivation as tree or other complex object, this implementation stores only the kind of MG deduction done to derive a given chart entry, and pointers to the antecedent chart entries. In ML, pointers are recreated as pairs of integers (address below) where the first member is an array index in the chart, and the second member is a linear position in the list at that array cell. The main role of type op is to name derivation tree branches.
type address = int * int

type op = R1 of address * address
| Ladj1 of address * address
| Radj1 of address * address
| Ladj2 of address * address
| Radj2 of address * address
| R1right of address * address
| R1left of address * address
| R1hopp of address * address
| R1hoplef of address * address
| R2 of address * address
| R3 of address * address
| R3right of address * address
| R3left of address * address
| R3hopp of address * address
| R3hoplef of address * address
| V1 of address
| V2 of address

module OrderedOp : (Set.OrderedType with type t=op) =
struct
t = op
let compare = compare
end

module BackpointerSet = Set.Make(OrderedOp)

type entry = {item: item
  self.address;
  mutable backpointers:BackpointerSet.t}

The record type for a chart entry, shown above, uses a version of the item type developed step-by-step in section A.2. Chart entries also must know their own address (the self field) and keep an alterable set of ops called the backpointer set. The standard library’s implementation of sets uses balanced binary trees in a way that makes union a relatively cheap operation.
A.4 The agenda

The agenda is a temporary holding area for items that are the result of deduction. They have not yet been entered into the chart, nor has their ability to derive further items been yet examined. In this implementation, the agenda is a queue, which leads to a breadth-first exploration strategy. It could equally be a stack (depth-first) or a priority queue (best-first).

Non-ground items complicate what would ordinarily be a simple redundancy check at enqueuing time, and make it impossible to use a stack queue component. The redundancy check is important for avoiding what Shieber, Schabes, and Pereira (1995) call an “avalanche of spurious overhead” (page 30). However, if items have variables in them the redundancy check has to be generalized to a subsumption check to catch the cases where an existing item covers a to-be-queued item in virtue of being strictly more general. While subsumption is easily derived from unification (section A.5) it would be wasteful to attempt it on each agenda entry. Rather, only the set of agenda entries that are likely to subsume an about-to-be-queued item should be checked.

To quickly extract the set of likely subsumers, the agenda is indexed by first feature, just like the chart. Unlike the chart, however, entries are sometimes removed from the queue, so these indices are not monotonic. The generic data structure is an “indexedQueue” where queue elements come equipped with a classification function assigning them to exactly one integer-labeled category. The queue itself is implemented as a circular buffer in an array named content that either contains Some entry of type α or has the value None (type 'a option). Pointers into content mark the head and tail of the queue, defining used and unused array slots. If more slots are needed a bigger array is allocated and the occupants are transferred to their new homes.
type 'a indexedqueue = {
  mutable byclass : IntListSet.t array;
  classify : 'a -> int;
  mutable head : int;
  mutable tail : int;
  mutable content : 'a option array;
}

let retrieve k q = if k < 0 || k >= (Array.length q.content) then raise Impossible_class
else List.map (function x -> match (Array.get q.content x) with
    Some v -> v
  | None -> raise Uninitialized)
(IntListSet.elements q.byclass.(k))

A.5 Unification

Unification allows the deduction-rule writer great freedom from the details of specifying exactly how parser items are matched. He or she need only state the conditions that matching inferences satisfy. The design of the implemented program reflects the ascription of a very high priority to this freedom, even at the expense of efficiency.

2Pointers into the content array and class labels are both integers. In the intended API the programmer never sees this confusing similarity, working only with the classification function classify and accessing the queue through the usual functions create, add, pop along with the quick lookup facility retrieve.
In particular, the program uses the unification-by-transformation algorithm given in section 4.7 of Baader and Nipkow (1998). It operates over a simple term data type in which everything is either a variable (V) or a headed functor (T).

\[
\text{type term} = \text{V of string} | \text{T of string} * \text{term list}
\]

For each parsing inference, events corresponding to the pseudocode in figure A.1 occur.

\[
a \text{trigger} \text{ item } x \text{ is popped off the queue}
\]

\[
\text{for each kind of parsing inference } I \text{ do}
\text{ retrieve from the chart all items whose first feature could cancel off the first feature of } x
\]

\[
\text{for all such matching } \text{anti-trigger} \text{ items } y \text{ do}
\text{ convert } x \text{ and } y \text{ to the Baader & Nipkow term type}
\text{ attempt to unify } x_{b\&n} \text{ and } y_{b\&n} \text{ and a pre-converted version of } I_{b\&n}
\text{ if such unification is successful and satisfies the Shortest Move Constraint (Stabler, 1997) then}
\text{ return Some consequent of } I \text{ converted back to a compact item}
\text{ else}
\text{ return None}
\text{ end if}
\text{ end for}
\text{ end for}
\]

Figure A.1: How unification is used for each case of parsing-as-deduction

The unification function actually used departs from Baader and Nipkow in two ways. First, following Prolog, the occurs-check is omitted. Since no circular terms arise parsing MGs, this is entirely innocuous. Second, functors having the special name append are treated specially; in traversing an append functor, the unifier assumes both arguments are lists, and uses underlying ML list primitives to make a new list using only a dot-headed (.) terms. This exposes the power (in the Prolog implementation) of being able to establish arbitrary Prolog side conditions for a deduction rule. In ML, the implementation is much less elegant because it uses multiple types for what are intuitively the same object. For example, both FLists below are treated the same way by the unification function:

\[
\text{FCons ((Feature.Select (Feature.Const "foo")),(FEmpty));}
\text{FAppend (FEmpty, ( FCons ((Feature.Select (Feature.Const "foo")),(FEmpty)))) ;}
\]

The only application of append in the program is to encode this particularly powerful type of side condition in the statement of the deduction rule. append terms are always eliminated before being passed back as the result of a rule application.

These results of attempting to apply each kind of deduction rule to the trigger item are collected in a list which is filtered by the subsumption testing algorithm in figure A.2.
This ensures that the agenda is kept as small as possible.

for all new items $u_{\text{new}}$ generated by algorithm A.1 do
    retrieve queue items $v_{\text{existing}}$ sharing the same first feature with $u_{\text{new}}$
    for all $v_{\text{existing}}$ do
        if $v_{\text{existing}}$ subsumes $u_{\text{new}}$ then
            $v_{\text{existing}}$.back pointers += $u_{\text{new}}$.back pointers \{accumulate $u$’s derivations in $v$\}
        else
            otherwise put $u_{\text{new}}$ on the queue
        end if
    end for
end for

Figure A.2: Subsumption testing in the queue

The Baader and Nipkow presentation was sufficiently clear that removing the occurs check and adding special handling of append-terms was straightforward. A more sophisticated approach might unify term graphs rather than concrete terms, as done here. Since MG items don’t typically have repeated subterms, there is no systematic structure to be shared and likely little to be gained.

A.6 Speed

The biggest speed penalty is failed unification attempts. To avoid these, guards have been put in that check for string position equality in arguments $x$ and $y$ to a deduction rule. For instance, in the second case of merge, the left boundary of $x$’s specifier must line up with the right boundary of $y$’s complement. ML code for a guard testing this condition is given below.

(* a weak filter on compatibility *)
let same p1 p2 = match (p1,p2) with
    PositionVar _ _ -> true (* if either is a variable, they have a *)
| (PositionVar _ _) -> true (* chance of unifying *)
| (Position x, Position y) -> x=y (* must match *)

let r2 compat = function
    (sl1,sl1,hl1,hr1,cl1,cr1, true,f1):_ _
    (sl2,sl2,hl2,hr2,cl2,cr2, f2):_ _ ->
        (same sl1 cr2) && (same sr2 hl2) && (same hr2 cl2)
| _ _ -> false

Applying the grammar described in appendix 4 to a test suite of 24 sentences from Keenan
and Hawkins (1987)\textsuperscript{3} about thirty-six percent of unifications actually succeed\textsuperscript{1} (figure A.3).

<table>
<thead>
<tr>
<th>average attempted (n=24)</th>
<th>max %</th>
<th>min %</th>
<th>average %</th>
</tr>
</thead>
<tbody>
<tr>
<td>9323</td>
<td>49.1</td>
<td>17.6</td>
<td>36.1</td>
</tr>
</tbody>
</table>

Figure A.3: Unification success rates

Figure A.4 compares the time required to find derivations for sentences in the Accessibility Hierarchy test suite. In all cases the machine used is an Apple Macintosh PowerBook G4 running at 500 MHz. The Prolog column reports timing using Sicstus Prolog 3.8.2; results in the ML column depend on the Objective Caml 3.06 native code compiler. Note that the Prolog runtimes are in minutes whereas the ML runtimes are in seconds. The latter figure neglects a brief initialization period of at most 0.1 seconds. On average, the ML version completes in $\frac{1}{160}$ the time of the Prolog version.

A.7 Conclusion

Shieber, Schabes, and Pereira (1995) showed how the natural advantages of Prolog lead to a clean implementation of even complicated parsing algorithms. The new implementation described here demonstrates that the natural advantages of ML can yield a faster parser while largely retaining the clarity and extensibility of the Prolog version.

\textsuperscript{3}The members of this test suite are supposed to be the output of morphological analysis (e.g. Beesley and Karttunen (2003)) that would resolve words like was into verbal root be and past tense marker -ed. The presupposition of such analysis is most evident in tense, but also shows up with WH-words. For instance, the grammar views whose as a morphological combination of who with the genitive suffix s.

\textsuperscript{4}These numbers are a conservative estimate in that they count as failures unifications that technically succeeded but whose result failed the Shortest Move Constraint.
<table>
<thead>
<tr>
<th>sentence</th>
<th>Prolog (mins)</th>
<th>ML (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>they have -ed forget -en that the boy who tell -ed the story be -ed so young</td>
<td>78.79</td>
<td>35.66</td>
</tr>
<tr>
<td>the fact that the girl who pay -ed for the ticket be -ed so poor doesn't matter</td>
<td>17.03</td>
<td>12.15</td>
</tr>
<tr>
<td>I know that the girl who get -ed the right answer be -ed clever</td>
<td>1.44</td>
<td>2.33</td>
</tr>
<tr>
<td>he remember -ed that the man who sell -ed the house leave -ed the town</td>
<td>272.39</td>
<td>107.16</td>
</tr>
<tr>
<td>they have -ed forget -en that the letter which Dick write -ed yesterday be -ed long</td>
<td>0.91</td>
<td>1.72</td>
</tr>
<tr>
<td>the fact that the cat which David show -ed to the man like -ed eggs be -ed strange</td>
<td>439.58</td>
<td>181.51</td>
</tr>
<tr>
<td>I know that the dog which Penny buy -ed today be -ed very gentle</td>
<td>0.481</td>
<td>1.08</td>
</tr>
<tr>
<td>he remember -ed that the sweet -ed which David give -ed Sally be -ed a treat</td>
<td>7729.29</td>
<td>1402.74</td>
</tr>
<tr>
<td>they have -ed forget -en that the man who Ann give -ed the present to be -ed old</td>
<td>196.14</td>
<td>79.57</td>
</tr>
<tr>
<td>the fact that the boy who Paul sell -ed the book to hate -ed reading be -ed strange</td>
<td>1734.36</td>
<td>626.94</td>
</tr>
<tr>
<td>I know that the man who Stephen explain -ed the accident to be -ed kind</td>
<td>127.05</td>
<td>51.22</td>
</tr>
<tr>
<td>he remember -ed that the dog which Mary teach -ed the trick to be -ed clever</td>
<td>128.10</td>
<td>54.11</td>
</tr>
<tr>
<td>they have -ed forget -en that the box which Pat bring -ed the apple be -ed lost</td>
<td>11.08</td>
<td>8.32</td>
</tr>
<tr>
<td>the fact that the girl who Sue write -ed the story with be -ed proud doesn't matter</td>
<td>3.38</td>
<td>4.12</td>
</tr>
<tr>
<td>I know that the ship which my uncle take -ed Joe on be -ed interesting</td>
<td>1.42</td>
<td>2.28</td>
</tr>
<tr>
<td>he remember -ed that the food which Chris pay -ed the bill for be -ed cheap</td>
<td>1.57</td>
<td>2.56</td>
</tr>
<tr>
<td>they have -ed forget -en that the girl who s friend buy -ed the cake be -ed wait -ing</td>
<td>5.43</td>
<td>5.13</td>
</tr>
<tr>
<td>the fact that the boy who s brother tell -ed lies be -ed always honest surprise -ed us</td>
<td>590.30</td>
<td>226.51</td>
</tr>
<tr>
<td>I know that the boy who s father sell -ed the dog be -ed very sad</td>
<td>211.90</td>
<td>69.44</td>
</tr>
<tr>
<td>he remember -ed that the girl who s mother send -ed the clothes be -ed come -ed too late</td>
<td>3.40</td>
<td>3.92</td>
</tr>
<tr>
<td>they have -ed forget -en that the man who s house Patrick buy -ed be -ed so ill</td>
<td>1.68</td>
<td>2.41</td>
</tr>
<tr>
<td>the fact that the sailor who s ship Jim take -ed have -ed one leg be -ed important</td>
<td>21.30</td>
<td>14.92</td>
</tr>
<tr>
<td>I know that the woman who s car Jenny sell -ed be -ed very angry</td>
<td>64.45</td>
<td>23.12</td>
</tr>
<tr>
<td>he remember -ed that the girl who s picture Clare show -ed us be -ed pretty</td>
<td>44.14</td>
<td>20.14</td>
</tr>
</tbody>
</table>

Figure A.4: Parsing sentences in Accessibility Hierarchy test suite
References


Jurafsky, Dan and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice-Hall.


Vita

John Tracy Hale was born in Tyrone, Pennsylvania on September 23rd, 1976. He grew up in State College, PA and graduated from State College Area High School in 1994. He received an Sc.B. in Cognitive Science and Computer Science from Brown University in 1998, writing an honors thesis in the area of statistical natural language processing. In 2001 he received an M.A. in Cognitive Science from the Johns Hopkins University. His M.A. thesis concerned connectionist parsing networks. He has been a visiting student at the University of California, Los Angeles and the Massachusetts Institute of Technology. Following completion of his Ph.D. he will begin an appointment as an Assistant Professor in the Department of Languages and Linguistics at Michigan State University.