

Sabrina Gerth

## Memory Limitations in Sentence Comprehension

A Structural-based Complexity Metric  
of Processing Difficulty







Potsdam Cognitive Science Series | 6

Sabrina Gerth

# **Memory Limitations in Sentence Comprehension**

A Structural-based Complexity Metric  
of Processing Difficulty

Universitätsverlag Potsdam

## **Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.dnb.de/> abrufbar.

## **Universitätsverlag Potsdam 2015**

<http://verlag.ub.uni-potsdam.de/>

Am Neuen Palais 10, 14469 Potsdam  
Tel.: +49 (0)331 977 2533 / Fax: 2292  
E-Mail: [verlag@uni-potsdam.de](mailto:verlag@uni-potsdam.de)

Die Schriftenreihe Potsdam Cognitive Science Series wird herausgegeben von Johannes Haack, Dr. Christiane Wotschack und Dr. Michael Dambacher

Zugl.: Potsdam, Univ., Diss., 2014  
Erster Gutachter: Prof. Dr. Shравan Vasishth  
Zweiter Gutachter: Prof. Dr. Gregory Kobele  
Tag der mündlichen Verteidigung: 20. November 2014

ISSN (print) 2190-4545  
ISSN (online) 2190-4553

Das Manuskript ist urheberrechtlich geschützt.

Online veröffentlicht auf dem Publikationsserver der Universität Potsdam  
URN <urn:nbn:de:kobv:517-opus4-71554>  
<http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-71554>

Zugleich gedruckt erschienen im Universitätsverlag Potsdam  
ISBN 978-3-86956-321-3

## Acknowledgements

Completing a PhD is quite a long journey and I would not have been able to complete it without the help and support of countless people. I would like to sincerely thank the many who helped me along the way. First on the list is my supervisor and guide, Shravan Vasishth. Shravan pushed me out of my comfort zone into my learning zone numerous times. He taught me to broaden my perspective on linguistics and was always ready with a good advice. His knowledge motivated me to try to find a good answer to his questions. Second on the list is Greg Kobele. Greg's coolness prevented many moments of panic. Without his enthusiasm, knowledge and support this thesis would have never been possible. I am grateful that I had the chance to work with both of you for all these years. I would also like to thank the rest of my committee: Frank Burchert, Guido Nottbusch and Jana Häussler. Special thanks go to Jana who took the time to understand this work and offered insightful comments.

This journey began with the encouragement of Peter beim Graben who introduced me to minimalist grammars and motivated me to broaden my perspective to an interdisciplinary approach on the topic. I am indebted to Heiner Drenhaus, Antje Saueremann, Ruben van de Vijver, Dinah Baer-Henney and Anja Gollrad who persuaded me to continue with this work despite any difficulties along the way. I thank you for your comments, suggestions and for every time you had a sympathetic ear and cheered me up.

This work has been shaped by my collaborations with John Hale and Marisa Ferrara Boston. John took the time to understand this work and offered helpful suggestions and advice. Marisa is the best conference buddy one can ask for. We enjoyed invaluable heart-to-hearts and she helped me to steady the steering wheel on the rough sea.

What would this journey be without my friends? This work benefited greatly from suggestions of Titus von der Malsburg, Pavel Logačev and Peter Baumann. I would like to thank my academic fellows: Felix Engelmann, Umesh Patil, Rukshin Shaher and Lena Jäger. Special thanks go to Vera Heyer, Clare Patterson (Thank you for proof-reading and spotting all my errors!), Gloria-Mona Knospe, Esther Schott, my dear friend Eliette and my second family, Julia and Yariv Festman and their wonderful children. Thank you all for giving me strength and reminding me of what really counts in life outside of science.

Most importantly, I want to thank my family, especially my parents, for their unfailing support, encouragement and love.



## Abstract

This dissertation addresses the question of how linguistic structures can be represented in working memory. We propose a memory-based computational model that derives offline and online complexity profiles in terms of a top-down parser for minimalist grammars (Stabler, 2011). The complexity metric reflects the amount of time an item is stored in memory. The presented architecture links grammatical representations stored in memory directly to the cognitive behavior by deriving predictions about sentence processing difficulty.

Results from five different sentence comprehension experiments were used to evaluate the model's assumptions about memory limitations. The predictions of the complexity metric were compared to the locality (integration and storage) cost metric of Dependency Locality Theory (Gibson, 2000). Both metrics make comparable offline and online predictions for four of the five phenomena. The key difference between the two metrics is that the proposed complexity metric accounts for the structural complexity of intervening material. In contrast, DLT's integration cost metric considers the number of discourse referents, not the syntactic structural complexity.

We conclude that the syntactic analysis plays a significant role in memory requirements of parsing. An incremental top-down parser based on a grammar formalism easily computes offline and online complexity profiles, which can be used to derive predictions about sentence processing difficulty.



# Zusammenfassung

Diese Dissertation beschäftigt sich mit der Frage wie linguistische Strukturen im Arbeitsgedächtnis dargestellt werden. Es wurde ein gedächtnisbasiertes Computermodell entwickelt, das offline und online Komplexitätsprofile für bestimmte Satzstrukturen erstellt. Diese Komplexitätsprofile basieren auf den Berechnungen eines top-down Parsers für minimalistische Grammatiken (Stabler, 2011). Die Komplexitätsprofile spiegeln die Zeit wider, die ein Item im Gedächtnis oder Speicher gehalten wird. Das entwickelte Modell verbindet grammatische Repräsentationen, die im Speicher gehalten werden, direkt mit dem kognitiven menschlichen Verhalten in Sprachverarbeitungsexperimenten indem es Vorhersagen über Satzverarbeitungsschwierigkeiten trifft.

Die Ergebnisse von fünf Satzverarbeitungsexperimenten wurden verwendet, um die Vorhersagen des Modells über die Begrenzungen des Gedächtnisses zu evaluieren. Die Vorhersagen der Komplexitätsmetrik wurden mit den Vorhersagen einer in der Psycholinguistik etablierten Komplexitätsmetrik (Dependency Locality Theory, DLT (Gibson, 2000)) verglichen. Beide Metriken machen vergleichbare offline und online Vorhersagen für vier der fünf Experimente. Der Unterschied zwischen beiden Metriken ist, dass die in dieser Dissertation erarbeitete Komplexitätsmetrik die strukturelle Komplexität der Satzstruktur mit einbezieht. Im Gegensatz zu DLTs Integrationsmetrik, die die Komplexität eines Satzes lediglich anhand der Anzahl der Wörter misst.

Diese Dissertation zeigt, dass ein inkrementeller Top-down Parser, der auf einem Grammatikformalismus basiert, offline und online Komplexitätsprofile berechnen kann, die verwendet werden können, um Vorhersagen über Satzverarbeitungsschwierigkeiten zu treffen.



# Contents

<b>Acknowledgements</b> . . . . .	v
<b>Abstract</b> . . . . .	vii
<b>Zusammenfassung</b> . . . . .	ix
<b>Contents</b> . . . . .	xi
<b>List of Tables</b> . . . . .	xv
<b>List of Figures</b> . . . . .	xvii
<b>1 Introduction</b> . . . . .	1
1.1 The Performance and Competence Debate . . . . .	2
1.2 The Role of Working Memory . . . . .	5
1.3 A Theory of Structural Complexity . . . . .	7
1.4 The Structure of this Thesis . . . . .	8
<b>2 Previous Work: Sentence Comprehension</b> . . . . .	9
2.1 Syntactic Ambiguity Resolution . . . . .	10
2.1.1 Canonical Sentoid Strategy . . . . .	10
2.1.2 Late Closure and Minimal Attachment . . . . .	12
2.1.3 Good-enough Approach . . . . .	12
2.1.4 Unrestricted Race Model . . . . .	13
2.2 Processing Overload and Parsing Breakdown . . . . .	15
2.2.1 Grammaticality Illusion – the Missing VP Ef- fect . . . . .	17
2.2.2 Complexity Measures: Previous Work . . . . .	18
2.3 Cognitive architectures . . . . .	22
2.3.1 Memory-based Theories . . . . .	22

2.3.2	Expectation-based Theories . . . . .	29
2.3.3	Memory-based and Experience-based Theories Combined . . . . .	33
<b>3</b>	<b>The Nature of the Human Parser . . . . .</b>	<b>37</b>
3.1	Derivational Theory of Complexity . . . . .	38
3.2	Structural Search of the Parser . . . . .	39
3.2.1	Filler-Gap Dependencies . . . . .	40
3.2.2	Licensing of Negative Polarity Items . . . . .	52
3.3	Structurally-mediated Reactivation . . . . .	54
3.3.1	Pronoun and Reflexive Resolution . . . . .	54
3.4	Computational Models of Structural-based Approaches . . . . .	60
3.4.1	ACT-R . . . . .	60
3.4.2	The Unification Space Model . . . . .	62
3.4.3	The Prediction Theory . . . . .	64
<b>4</b>	<b>The Parsing Algorithm . . . . .</b>	<b>67</b>
4.1	The Minimalist Grammar . . . . .	67
4.1.1	Structure Building Operations . . . . .	69
4.1.2	Derivation Trees . . . . .	75
4.2	Parsing Strategies . . . . .	76
4.2.1	Bottom-up and Top-down Parsing . . . . .	76
4.2.2	Left-corner Parsing . . . . .	77
4.3	The Components of the Implemented Minimalist Top-down Parser . . . . .	79
4.3.1	The Grammar Fragment . . . . .	80
4.3.2	The Oracle of the Parser . . . . .	83
4.3.3	The Top-down Parsing Algorithm . . . . .	84
4.3.4	The Complexity Metric . . . . .	95
<b>5</b>	<b>Results . . . . .</b>	<b>99</b>
5.1	Modeling the Offline Data . . . . .	101
5.1.1	Nesting Constructions: SC/RC vs. RC/SC Embedding . . . . .	101

5.1.2	Center-embedded vs. Right-branching Structures . . . . .	102
5.1.3	Subject-modified vs. Object-modified SRC or ORC . . . . .	106
5.2	Modeling the Online Data . . . . .	108
5.2.1	Subject-extracted vs. Object-extracted Relative Clauses . . . . .	108
5.2.2	Locality Effects . . . . .	112
5.2.3	Complex NP Including a PP Modifier vs. RC Intervening . . . . .	122
<b>6</b>	<b>General Discussion . . . . .</b>	<b>127</b>
6.1	Conclusions . . . . .	127
6.2	Future Work . . . . .	128
<b>7</b>	<b>The Grammar Fragment . . . . .</b>	<b>131</b>
<b>8</b>	<b>Result Tables for the Offline Data . . . . .</b>	<b>135</b>
	<b>Bibliography . . . . .</b>	<b>141</b>





## List of Tables

5.1	Comparison of data and predictions of the two complexity metrics . . . . .	100
5.2	Comparison of data and predictions of maxDLT and average tenure for nesting constructions SC/RC vs. RC/SC . . . . .	102
5.3	Comparison of data and predictions of maxDLT and average tenure for center-embedded vs. right-branching structures . . . . .	104
5.4	Comparison of data and predictions of DLT and tenure for the embedded verbs of the two-level center-embedded sentence . . . . .	105
5.5	Comparison of data and predictions of maxDLT and average tenure for subject-modified vs. object-modified SRC or ORC . . . . .	107
5.6	Word-by-word predictions for data from Experiment 1, Grodner and Gibson (2005) . . . . .	110
5.7	Comparison of data and predictions of DLT and tenure for the embedded verbs . . . . .	112
5.8	Word-by-word predictions for data from Experiment 2, Grodner and Gibson (2005) . . . . .	115
5.9	Comparison of data and predictions of DLT and tenure for the embedded verb <i>supervised</i> . . . . .	116
5.10	Word-by-word predictions for data from Experiment 2, Grodner and Gibson (2005) . . . . .	117
5.11	Comparison of data and predictions of DLT and tenure for the embedded verbs <i>supervised</i> and <i>scolded</i> . . . . .	118
5.12	Comparison of data fit for DLT and tenure . . . . .	121

5.13	Word-by-word predictions for data from Gibson and Warren (2004) . . . . .	124
5.14	Comparison of data and predictions by tenure for the verb phrase <i>had pleased</i> . . . . .	126
8.1	Word-by-word predictions for nesting constructions RC/SC vs. SC/RC . . . . .	136
8.2	Word-by-word predictions for center-embedded vs. right-branching structures . . . . .	137
8.3	Word-by-word predictions for subject-modified vs. object-modified SRC or ORC . . . . .	138
8.4	Word-by-word predictions for subject-modified vs. object-modified SRC or ORC . . . . .	139

## List of Figures

3.1	Phrase structure trees of SRs and ORs . . . . .	47
3.2	Lexical elements in ACT-R's memory . . . . .	61
3.3	Lexical frames in the Unification Space model . . . . .	62
3.4	Elementary trees in the lexicon of PLTAG . . . . .	64
4.1	A tree built by a minimalist grammar . . . . .	68
4.2	A tree violating the Shortest Movement Constraint . . . . .	69
4.3	Merge <i>the</i> and <i>employee</i> . . . . .	70
4.4	Merge <i>hired</i> and <i>the employee</i> . . . . .	70
4.5	Move <i>the employee</i> . . . . .	71
4.6	Merge <i>the manager</i> . . . . .	72
4.7	Move <i>the manager</i> . . . . .	73
4.8	Merge $\epsilon$ to <i>the manager the employee hired</i> . . . . .	74
4.9	Derivation tree of <i>The manager hired the employee</i> . . . . .	75
4.10	Move <i>manager</i> into specifier position to the left of <i>who</i> . . . . .	81
4.11	Left incorporation of <i>from</i> as head of prepositional phrase <i>from the clinic</i> . . . . .	82
4.12	Affix hopping to combine tense affixes and verb stems . . . . .	83
4.13	Derivation tree for <i>The manager will hire the employee</i> . . . . .	94
4.14	Derivation tree with tenure counts for <i>The manager will hire the employee</i> . . . . .	97
5.1	Comparison of data from Grodner and Gibson (2005) and the models' predictions for the SRC and the ORC . . . . .	111
5.2	Comparison of data for Expt. 2 (matrix conditions) by Grodner and Gibson (2005) and the models' predictions for the locality examples . . . . .	119

5.3	Comparison of data for Expt. 2 (embedded conditions) by Grodner and Gibson (2005) and the models' predictions for the locality examples . . . . .	120
5.4	Comparison of data and the predictions of tenure for examples containing an intervening complex NP and PP vs. a RC . . . . .	125

# 1 Introduction

The ability to understand a sentence in real-time draws on a very complex cognitive mechanism. Over the past decades, researchers have come up with a number of questions and theories to address sentence comprehension. What is the mental representation of comprehending a sentence? Why are some sentences more difficult to understand than others? What are the algorithms and constraints that we need to apply during sentence comprehension?

In order to answer these questions we need to understand the cognitive mechanism that underlies sentence comprehension. Marr (1982) suggests that to fully understand a mechanism we need to understand the pieces it consists of and how these parts are linked. Marr (1982) defines three levels of information processing systems such as human cognitive systems. First, the *computational theory* describes the goal of the computation and what is actually computed in a rational analysis, for instance a grammatical theory. Second, the *representation and algorithm* defines how the computational theory is implemented and how the calculation works. The algorithm is the transformation of the input representation (e.g., words or constituents in a sentence) to the output (e.g., a grammatical syntactic structure) and it defines intermediate steps such as intermediate phrase-structure trees in a syntactic structure. The third level is the *implementation* which states how the algorithm and the representation are realized, for instance as neurons in the brain or intermediate states of the mind. The choice of algorithm is directly influenced by the type of hardware in which it will run.

If we apply Marr's vision of an information processing system to language, we assume that the garden-path theory is a computational theory which is represented in an algorithm and a

possible underlying mechanism (e.g., connectionism). This thesis attempts to represent the theory of processing complexity in sentence comprehension through an algorithm, namely an incremental parser based on an established grammar formalism. The algorithm is implemented in an architecture which derives syntactic complexity profiles based on limitations in working memory capacity for phenomena of processing complexity. Thereby, this work describes a direct relationship between grammatical representations and cognitive behavior by deriving predictions about sentence processing difficulty and comparing these predictions to existing empirical findings. We assume a single cognitive mechanism in which the grammar is an abstract description of the representations that the cognitive system builds during language processing (similar to other computationally inspired human parsing models (Hale, 2001; Lewis and Vasishth, 2005; Roark, 2001)).

The following three subsections will introduce (1) the role of grammar as an abstract description of what we know and how we behave during language processing, (2) the limits in capacity and time of working memory while comprehending language and (3) how sentence complexity can be defined in terms of structural distance.

### **1.1 The Performance and Competence Debate**

Schlesinger (1968) wrote “the human user of language incorporates a device which operates along the lines of a grammar proposed by theoretical linguists”. The grammar that Schlesinger (1968) had in mind was a Chomskyan grammar (Chomsky, 1957). Since this statement is very general, Schlesinger calls it a *meta-hypothesis* from which more specific *research hypotheses* can be derived. The syntactic theory of Chomsky (1957) assumed that knowing the syntax of a language comprised knowing the phrase structure rules which generate the underlying structures into sen-

tences (Townsend and Bever, 2001). Grammars define a mapping between the surface form of a sentence and its underlying meaning (Hawkins, 1994). The speaker must be capable of using this mapping in real-time without any interruptions to convey information rapidly and efficiently to the listener.

For many years linguists insisted that grammar is only a model of *competence*, i.e. what the speaker knows. Hence, grammar cannot reflect effects of *performance*, i.e. the speaker's behavior (Townsend and Bever, 2001). This argument was partly motivated by the fact that some sentences are grammatical but still notoriously difficult to understand. For instance, Miller and Chomsky (1963) observed that center-embedded sentences become increasingly difficult to comprehend with an increasing level of embedding. However, some center-embedded sentences are comprehensible (example taken from Townsend and Bever (2001)):

- (1) The reporter [everyone [I met] trusts] had predicted the coup.

Therefore, linguists concluded that there is a clear distinction between performance and competence.

The syntactic theory of Chomsky (1957) provided an answer to the question of what we know if we know a language (competence). We know the syntactic rules and the levels of representation they describe. The phrase structure rules define a *deep structure* of the sentence and syntactic transformations derive the *surface form* of the sentences from these deep structures. Additionally, Chomsky tried to answer the question about language performance. He linked the syntactic theory to language behavior such that every syntactic operation corresponded to a psychological process. It follows that the behavioral complexity of sentence processing is directly linked to the number of transformations that are necessary to construct the syntactic structure of the sentence. In other words, Chomsky directly linked the number of grammar operations to the perceived processing complexity.

Townsend and Bever (2001) also criticize the clear-cut distinction between competence and performance. In particular, they argue

against the view that competence has “no particular implication for mental structures” (Townsend and Bever, 2001, p. 27) and only performance can actually map behavior and cognitive processes. Townsend and Bever (2001) state that competence itself can also be mapped onto behavior, i.e. through grammaticality judgments that test the knowledge a speaker has about the language.

Another challenge, according to Townsend and Bever (2001), was to show that *deep structures* could be mapped onto behavior. How can grammatical rules, which represent only an intermediate state of the phrase structure representation, be mapped onto mental processes? The work by Miller and Chomsky (1963) linked linguistic knowledge (competence) and linguistic behavior (performance) in a very abstract way. One approach which attempted to formalize their ideas was the *Derivational Theory of Complexity* (DTC) proposed by Fodor and Garrett (1966). The DTC directly linked the complexity of a sentence to the number of grammatical operations applied to the structure (Fodor and Garrett, 1966). Interestingly, the transformational distance between sentences, e.g. an active compared to a passive sentence, predicted processing complexity in experiments where participants had to recall sentences, i.e. a passive question was more often recalled as an active question than as a passive one. However, the derivational hypothesis made wrong predictions for other constructions (see Chapter 3.1 for a detailed discussion).

Until now, the attempt to use grammar in a processing model to explain sentence comprehension difficulties has failed (Townsend and Bever, 2001). Grammar can only detail knowledge about language but not the specific kind of language behavior, since there are numerous models of language behavior and each has its own neurological and physical constraints. Nevertheless, more recently researchers have argued for a unified model of grammatical theory and language processing, simply considering them as different levels of abstraction (Lewis and Phillips, under review).



## 1.2 The Role of Working Memory

The concept of working memory and its limits are the key part of human cognition (Cowan, 2005). Working memory can be thought of as a system which actively maintains and processes a certain amount of information in a short time. Naturally, the human memory is *limited in capacity* and *time-bound* (Cowan, 2000; Frazier, 1979; McElree, 2006; Miller, 1956; Wagers, 2013). It is limited in capacity in the sense that only a certain amount of material can be stored simultaneously, and it is time-bound because stored material decays over time as long as it is not integrated into the sentence structure or reactivated (Gibson, 2000; Hawkins, 1994; Just and Carpenter, 1980, 1992; Lewis and Vasishth, 2005; Ricker et al., 2010).

In language processing we need to retain earlier parts of the message until they can be integrated into the structure. If we fail to memorize relevant information we will have difficulties comprehending the meaning of the sentence.

One can imagine that extremely long sentences with more than 50 words will overwhelm the limits of working memory. The more interesting sentences, though, are short syntactically complex constructions that are hard to process, e.g. center-embedded sentences. In earlier work in psycholinguistics working memory was regarded as a simple temporary buffer which was limited by its capacity (Cowan, 2000; Miller, 1956). Recent work revived the question of how structural information is actually encoded and retrieved from memory (Kobele et al., 2013; McElree, 2006; Wagers, 2013).

Human sentence comprehension works incrementally (Marslen-Wilson, 1973; Phillips, 2003; Tanenhaus et al., 1995). Syntactic representations are built online while comprehending a sentence and one does not wait until the end of the sentence to begin processing. Empirical results show that each word is incorporated into the syntactic structure as soon as it is read (Frazier and Fodor, 1978; Frazier, 1979; Just and Carpenter, 1980; Pickering et al., 1999). Additionally, syntactic structures are fully connected,

i.e. new words are integrated into the same syntactic structure (Levelt, 1970; Stabler, 1994; Sturt and Lombardo, 2005). Incremental processing involves the resolution of linguistic relations, i.e. dependencies. Dependencies can be built between noun phrases and verbs, pronouns or reflexives and their antecedents, gaps and fillers. Sometimes the human parser has to delay a specific structural analysis of the sentence for quite some time until relevant information, e.g. the right-hand element of a dependency, occurs in the sentence. During this search of the parser for relevant syntactic information, the processor might also exclude possible incoming information based on the bottom-up input, i.e. the currently processed words. Hence, information and partial syntactic structure has to be stored in memory until it gets reactivated and retrieved.

The idea that this thesis will pursue is that the information in working memory is stored in a hierarchical structure which facilitates storage. This is because the hierarchical structure, i.e. connected chunks of words, increases memory capacity limits (Cowan, 2005; Miller, 1956). Several words that can be grouped together might use up only one chunk in memory. If we exploit a certain pattern or structure in the material that we have to memorize then it might become easier to retrieve these chunks of information later. The implemented parsing algorithm will work incrementally from left-to-right to directly simulate human reading. In other words, the parser will not wait until all words are processed but starts building the syntactic structure right-away. The algorithm is based on a grammar formalism that derives the hierarchical structure of the processed words (for more details see Section 4). From this it will calculate syntactic complexity profiles that capture the time a constituent is stored in memory (the results are discussed in Section 5). This way, the key characteristics of language processing are captured by this algorithm: it works incrementally on connected structures and accounts for the capacity and time limit on working memory.

## 1.3 A Theory of Structural Complexity

*Processing complexity* is one of the crucial topics in sentence comprehension. The concept of processing complexity relies on various linguistic and computational approaches to represent and build sentence structure (Frazier and Fodor, 1978; Gibson, 1998; Hawkins, 1994; Kimball, 1973; Lewis, 1996). The concept of processing complexity is based on the fact that parsing a sentence requires the assignment of resources, a process which should be observable not only in ambiguous structures but also in complex constructions, e.g. center-embeddings, left- and right-branching constructions, and heavy constituents (Sekerina, 2003).

One question in the sentence processing literature is: What is the correspondence between the syntactic structure and the perceptual complexity of a sentence?

According to Hawkins (1994), previous research defined difficulty in terms of certain structures that are difficult to comprehend and others that are not. Then, linguists agreed that difficulty should rather be defined in a more gradient way in terms of degrees of processing difficulty. On the one hand, according to resource-based approaches of complexity, the difficulty of integrating a new item into the current syntactic structure depends on the length of the material intervening between the item and its head (Gibson, 1998). On the other hand, a number of studies claim a direct relationship between the number of grammar rules to build the syntactic structure of a sentence and variations in performance, i.e. processing difficulties in sentence comprehension (e.g., Miller and Chomsky, 1963; Frazier, 1985). In other words, the structural complexity of a sentence is measured by counting syntactic nodes in the last case. However, by simply counting nodes one would undermine the syntactic relationship between them such as dominance or c-command (Hawkins, 1994). In order to determine the syntactic dependencies in a more fine-grained manner, a mechanism, e.g. a parser, based on an established grammar formalism is needed. Grammars are

sensitive to processing load in terms of the number of necessary syntactic operations to produce the syntactic structure of a sentence. Hence, using a grammar formalism as the basis for the calculation of syntactic complexity seems straightforward and natural. Moreover, taking into consideration the processing or memory cost of intermediate structures that are not yet integrated into the global sentence structure would add a fine-grained calculation to the otherwise rather coarse-grained method of counting intervening words or discourse referents.

### **1.4 The Structure of this Thesis**

The thesis is organized as follows. Chapter 2 outlines previous research on syntactic ambiguity resolution, processing overload and cognitive architectures in the context of memory-based and expectation-based theories. The focus is here on phenomena with processing overload and implemented architectures relating processing difficulties in sentence comprehension to limits in working memory capacities. Chapter 3 reviews previous research on the human parser as the language processing system. The chapter outlines theories and empirical evidence contrasting structural-based and linear-based approaches to explain syntactic complexity. This thesis argues that structural-based complexity metrics are more fine-grained and hence more powerful in explaining observed processing difficulties in sentence comprehension. Chapter 4 describes the implemented parsing algorithm which is based on an established grammar formalism. The parser works incrementally on connected structures and derives offline and online complexity profiles for phenomena of processing overload. These structural-based complexity metrics reflect the amount of time an item is stored in memory. Chapter 5 compares the predictions of a locality cost metric (DLT, Gibson (2000)) to the predictions of the implemented algorithm for empirical data from acceptability judgments (offline) and reading times data (online). Finally, chapter 6 concludes and discusses suggestions for further research.

## 2 Previous Work: Sentence Comprehension

Most of the time language processing is easy, but under certain circumstances sentence comprehension becomes difficult. Therefore a plausible model of sentence comprehension must account for both possibilities, the cases in which processing works without any conscious difficulty, and cases in which processing is hard or not possible at all.

Two phenomena of processing difficulty in sentence comprehension have been studied to a great extent in psycholinguistics: *Syntactic Ambiguity Resolution* and *Processing Overload Effects* (see Van Gompel and Pickering, 2007, for a review).

This chapter will outline previous theories of both phenomena. Firstly, in Section 2.1 *Syntactic Ambiguity Resolution* theories regarding the garden-path effect are briefly discussed. The section begins with Bever's (1970) "Canonical Sentoid Strategy", then discusses Frazier's (1979) "Late Closure" and "Minimal Attachment" proposals, which are followed by Ferreira's (2003) "Good-enough Approach" and finally van Gompel, Pickering, Pearson and Liversedge's (2005) "Unrestricted Race Model". Section 2.2 *Processing Overload and Parsing Breakdown* summarizes previous work in the area of complexity phenomena. Several complexity metrics are summarized: Yngve's (1960) "Depth Metric", Miller and Chomsky's (1963) "Global Nonterminal Ratio", Frazier's (1979) "Local Nonterminal Count" and Kimball's (1973) "Principle of Two Sentences". Finally, the third part of this chapter compares two distinctive directions of cognitive architectures in the recent psycholinguistic literature: *Memory-Based* and *Expectation-Based* Theories.

## 2.1 Syntactic Ambiguity Resolution

Garden-path effects are one example of ambiguity resolution. For instance if a sentence has a local ambiguity with a preference for one reading, but it turns out that the alternative reading is the correct one, then this sentence causes a garden-path effect and has to be reanalyzed to the correct interpretation. In some cases the processing difficulty does not reach a conscious level (*weak garden-path effect*), in other situations the difficulty causes a conscious confusion and reanalysis is necessary (*strong garden-path effect*), and in a third case it is not possible to fully understand the sentence (*parsing breakdown*). Empirical evidence for the garden-path effect comes from a great number of psycholinguistic experiments (Bever, 1970; Ferreira, 2003; Ferreira et al., 2002; Ferreira and Clifton, 1986; Frazier and Fodor, 1978; Frazier and Rayner, 1982, 1987; MacDonald et al., 1994; Mitchell, 1994; Tanenhaus and Trueswell, 1995; Trueswell et al., 1994; Van Gompel et al., 2005, among others).

### 2.1.1 Canonical Sentoid Strategy

One of the first and most famous examples of a garden-path sentence in the literature is the following sentence from Bever (1970):<sup>1</sup>

- (2) # The horse raced past the barn fell.

Sentence (2) is difficult because it is locally ambiguous between a main-verb and a reduced-relative reading of the word *raced*. This kind of ambiguity is often resolved in favor of the main clause analysis causing a garden-path effect when reading the final verb *fell*. Bever (1970) explained this effect by proposing several heuristics, one of which was the *Canonical Sentoid Strategy*. This heuristic hypothesizes that “the first clause [...] is the main clause unless the verb is marked as subordinate” (Bever, 1970, p. 294).

---

<sup>1</sup>Sentences that cause conscious processing difficulty are prefixed with the symbol ‘#’.

However, Gibson (1991) shows that Bever's heuristic cannot account for all garden-path effects. Local attachment preferences as shown in (3) cannot be explained by Bever's (1970) explanation.

- (3) # The patient persuaded the doctor that he was having trouble with to leave.

The complement clause *that he was having trouble with* is preferably attached locally to the verb *persuaded*, rather than the noun phrase *the doctor* which turns out to be the correct attachment site. This attachment preference causes a garden-path effect because the sentence cannot be parsed correctly if the complement clause is wrongly attached to the verb.

Another example for which Bever's heuristic makes the wrong predictions according to Gibson (1991) is an example from Frazier and Rayner (1987).

- (4) The warehouse fires destroyed all the buildings.  
(5) The warehouse fires a dozen employees each year.

The word *fires* is lexically ambiguous between a main verb and a noun reading. The preference for one or the other is not high enough to cause a garden-path effect and the local ambiguity is immediately resolved at the next word (Frazier and Rayner, 1987). Bever (1970) predicts a preference for the main verb reading which should cause processing difficulty for sentence (4).

Nevertheless, Bever's theory is one of the first testable theories that tried to explain garden-path effects (Gibson, 1991). Even though there are to date several examples that this theory cannot explain, it is the starting point for all recent work on human performance in linguistics.

### 2.1.2 Late Closure and Minimal Attachment

Frazier (1979) formulated the most frequently cited strategies in the processing literature of garden-path theory: *Late Closure* and *Minimal Attachment*.

Late Closure: "When possible attach incoming material into the phrase or clause currently being parsed." (Frazier, 1979, p. 33)

Minimal Attachment: "Attach incoming material into the phrase-marker being constructed using the fewest nodes consistent with the well-formedness rules of the language under analysis." (Frazier, 1979, p. 24)

These two principles account for a large amount of empirical evidence of garden-path effects and reading preferences, including (2) and (3) (Ferreira and Henderson, 1991; Frazier and Fodor, 1978; Frazier and Rayner, 1982; Mitchell and Holmes, 1985). But, Gibson (1991) criticizes the fact that the strategies have not been formalized to be testable with empirical data, i.e., using values that measure processing difficulty in online sentence processing.

### 2.1.3 Good-enough Approach

Ferreira (2003) proposed the *Good-Enough* approach (see also Ferreira et al., 2001, 2002). Sometimes reanalysis is not an all-or-none phenomenon. According to Ferreira (2003) the syntactic representation of a sentence can be *good enough* to lead to a satisfying interpretation even though it is not detailed enough to distinguish important differences, for instance, who is doing what to whom. In sentence (6) participants initially interpret *the baby* as the object of *dressed*. As soon as they read *spit up* they realize this interpretation is wrong and they have to reanalyze *the baby* to be the subject of *spit up*. It is assumed that participants will end up with the correct analysis of the sentence after reanalysis. But, when they are asked the question *Did Anna dress the baby?* participants mostly



answered incorrectly with *yes* (Ferreira et al., 2002). Ferreira takes this as evidence that the complete reanalysis of the syntactic structure failed and people ended up with interpreting *the baby* as both the object of *dressed* and the subject of *spit up*.

- (6) While Anna dressed the baby spit up on the bed.
- (7) While Bill hunted the deer (that was brown and graceful) ran into the woods.

Furthermore the distance between the head of the ambiguous phrase and the point of disambiguation matters for a successful reanalysis (Ferreira et al., 2002). If the distance is greater, people are committed to the wrong interpretation of the sentence for a longer time and the probability of a successful reanalysis decreases. Ferreira et al. (2002) manipulated the length of the ambiguous region as in (7) by adding more material (shown in brackets) and found that the greater the distance between the head of the ambiguous phrase (*deer* in (7)) and the disambiguation point (*ran*), the more difficult the reanalysis becomes.

Ferreira (2003) demonstrates that her approach also applies to sentences with non-canonical word orders like passive sentences (compared to active sentences and subject clefts) and object clefts (compared to subject clefts). Participants were both more accurate in making a oral decision between possible interpretations and faster in responding to active sentences and subject clefts. Ferreira concludes that participants have difficulty assigning thematic roles in an atypical order (e.g., passives, object clefts).

#### 2.1.4 Unrestricted Race Model

Van Gompel et al. (2005) suggest the *Unrestricted Race Model* as a ranked parallel model of sentence comprehension. They define the model as a variable-choice reanalysis model which adopts the syntactic analysis that wins the race of differently ranked analyses at each word. Only one analysis is pursued while the alternatives are abandoned before parsing the sentence is finished. Multiple

analyses at different stages of processing do not increase processing difficulty because each alternative draws upon independent processing resources.

If the ambiguity in the sentence is biased towards one interpretation then this analysis is facilitated and nearly always wins the race. If disambiguation results in a structure that is inconsistent with the structure built so far the processor has to reanalyze which leads to processing disruption. The processor uses syntactic information (e.g., word category) prior to non-syntactic information (e.g., semantic plausibility).

Van Gompel et al. (2005) tested two types of ambiguities: (1) relative clause (RC) attachment ambiguities, and (2) verb phrase (VP)/noun phrase (NP) attachment ambiguities. They contrasted a globally ambiguous condition with high and low attachment conditions for both types of ambiguities. They claim that the globally ambiguous condition is easier to process than the disambiguated sentences since the processor is not forced to reanalyze. However, as they state in the discussion section of the paper, participants did not need to resolve the ambiguity in the sentence to answer the comprehension questions correctly. The results of experiment 2 (Van Gompel et al., 2005, p. 293) show no reliable differences in any of the eye-movement measures between the globally ambiguous and the syntactically unambiguous condition. This could indicate that participants parsed both conditions in the same way which would point to a single analysis without any competition, and strictly speaking could not be taken as evidence against competition-based models.

## 2.2 Processing Overload and Parsing Breakdown

The second phenomenon under discussion in this chapter which contributes to processing difficulty is *Processing Overload or Parsing Breakdown* (Cowper, 1976; Frazier, 1985; Gibson, 1991, 1998, 2000; Gibson and Thomas, 1999; Kimball, 1973; Lewis, 1996; Miller and Chomsky, 1963; Vasishth et al., 2010; Yngve, 1960, among others). Sentences causing this type of processing difficulty differ from those causing garden-path effects because they are fully grammatical and no ambiguity has to be resolved. Multiply center-embedded structures are one example of structures that can cause processing overload (Bever, 1970; Cowper, 1976; Gibson, 1991, 1998; Gibson and Thomas, 1999; Kimball, 1973; Lewis, 1993; Miller and Chomsky, 1963; Stabler, 1994; Vasishth et al., 2010). Nearly all theories about parsing breakdown attribute processing difficulty to limitations of the computational resources of the language processor (Abney and Johnson, 1991; Chomsky, 1965; Kimball, 1973; Lewis, 1993; Rambow and Joshi, 1994; Stabler, 1994).

Before summarizing previous work on complexity metrics, a few important characteristics of processing overload phenomena will be listed.

### Independence of Ambiguity

As stated above, sentences causing parsing breakdown do not necessarily contain an ambiguity. Lewis (1993) demonstrates this fact with two examples, given below as (8) and (9). Here, a local ambiguity can be caused by omitting the complementizer *that* in center-embedded structures:

- (8) The cat the bird the mouse chased scared ran away.
- (9) The cat that the bird that the mouse chased scared ran away.

Even though sentence (9) contains the complementizer *that*, the structure is still unacceptable and hard to understand. Therefore,

having a local ambiguity in a structure without a complementizer is not an explanation for its unacceptability (Fodor and Garrett, 1967; Lewis, 1993).

### **Levels of Embedding**

Sentences with one level of embedding are judged as being grammatical in acceptability judgment tasks, while additional levels of embedding are rated ungrammatical and unacceptable (Blaubergs and Braine, 1974). Interestingly, the step from two to three levels of embedding is even wider than one to two, in the sense that three levels of embedding cannot be understood or parsed (*Parsing Breakdown*). This distinguishes sentences with parsing breakdown from garden-path structures. Even with the right training and explicit instructions participants still find center-embedded structures difficult to comprehend, while they are capable of learning how to parse garden-path sentences correctly (Blaubergs and Braine, 1974).

### **Independence of Length**

According to Blaubergs and Braine (1974), increasing the length of a sentence does not cause an increase in processing difficulty. They manipulated the length of center-embedded sentences from one level to five levels of embedding and compared these sentences to right-branching structures with similar meaning and equivalent length. The results confirm that parsing breakdown is not predictable based on the length of sentences.

### 2.2.1 Grammaticality Illusion – the Missing VP Effect

Grammatical illusions are an interesting phenomenon which can demonstrate online parsing processes. An interesting observation in this domain is called the *missing VP effect* (Frazier, 1985; Gibson and Thomas, 1999; Vasishth et al., 2010). Under certain circumstances readers overlook the fact that a center-embedded sentence is missing a syntactically obligatory VP. Gibson and Thomas (1999) examined sentences like the following in acceptability rating experiments:

- (10) The ancient manuscript [that the graduate student [who the new card catalog had confused a great deal] was studying in the library] was missing a page.
- (11) The ancient manuscript [that the graduate student [who the new card catalog had confused a great deal]] was missing a page.

In (10) all three VPs are present while in (11) the second VP *was studying* is omitted.<sup>2</sup> Their results confirmed that both sentences were rated equally as acceptable even though the second sentence was ungrammatical. Gibson and Thomas (1999) conclude that the predicted middle VP is forgotten due to capacity overload of the processor and this facilitates parsing afterwards because there are fewer items in memory.

Vasishth et al. (2010) replicated these results from the offline acceptability rating in several online task (self-paced reading and eye-tracking) experiments. Their findings show that omitting the middle VP facilitates the processing of doubly center-embedded sentences in English but not in German. Vasishth et al. (2010) conclude that the grammaticality illusion is not a language-universal phenomenon. It is rather dependent on language-specific factors such as head-finality and perhaps grammar rules governing punctuation in German.

---

<sup>2</sup>Gibson and Thomas (1999) also omitted the first and third VP to test which of the three VPs is forgotten, and determined it was the middle VP.

## 2.2.2 Complexity Measures: Previous Work

This section outlines several complexity metrics from the literature. Complexity metrics calculate the perceived complexity or processing difficulty of a syntactic structure. Most of the metrics define a limit of complexity. A structure with a complexity value that exceeds this limit is predicted to be unacceptable.

### 2.2.2.1 Yngve's depth metric

Yngve (1960) proposed one of the first complexity metrics. Although the metric was originally intended for sentence production, it can be used for parsing as well. Yngve's metric adopts a top-down depth-first technique on phrase structure rules by starting at the top-most phrase node and expanding the rules recursively from left to right one word at a time. The algorithm operates on a context-free grammar and uses a temporary memory to store nodes that need to be expanded. The measure of complexity counts how many categories are stored in memory at the point of generating a required node. The temporary memory is equivalent to a stack in which only the top-most item is accessible. Yngve (1960) limits memory size to seven items motivated by Miller's theory of short-term memory (Miller, 1956). This limit seems to work well for right-branching structures.

According to Gibson (1991), Yngve's metric correctly distinguishes between right-branching and center-embedded structures. This distinction is demonstrated with examples (12) and (13). Both sentences convey the same information, but (13) is easier to process than (12). According to Yngve's metric the maximal complexity of (12) occurs at the second and third determiner of the NPs *the woman* and *the dog* with four constituents in memory. At the second determiner there are one N, one S' and two VP nodes, and at the third determiner one N and three VP nodes.

(12) # The man that the woman that the dog bit saw likes fish.

(13) The man that was seen by the woman that the dog bit likes fish.

While the metric makes the right predictions for right-branching and center-embedded sentences, according to Kimball (1973) it is problematic for left-branching structures. The problems are not due to the metric itself but to the underlying top-down strategy (Kimball, 1973). The algorithm predicts that unbounded left-branching is unacceptable. This prediction might be correct for English which is predominantly right-branching, but it cannot explain predominantly left-branching languages, such as Japanese, in which speakers do not encounter the processing difficulty that would be predicted by Yngve's metric.

Despite these issues the complexity metric has some desirable features. It is implemented in a well-defined computational architecture, it makes clear and measurable predictions, and the limitations on the architecture are based on an independently developed theory (Lewis, 1993).

### 2.2.2.2 Miller and Chomsky's global nonterminal ratio

Miller and Chomsky (1963) proposed several measures to calculate complexity. One is the *degree of self-embedding* which presumes that increased levels of embedding leads to more difficulty in comprehension. However, as Lewis (1993) points out, Miller and Chomsky do not specify any particular limit for embedding and they kept the specification very abstract. Another complexity measure that Miller and Chomsky proposed is *structural complexity* calculated by "the ratio of the total number of nodes in the hierarchy to the number of terminal nodes" (Miller and Chomsky, 1963, p. 485). Gibson (1991) points out that the metric cannot predict correctly the asymmetry between center-embedded structures and acceptable structures, like left-branching and right-branching sentences. But, the measure was originally not intended to distinguish between these types of embeddings (Lewis, 1993). Even though the measures of complexity proposed by Miller and Chomsky (1963) are only descriptive, they provided a starting point for nearly all theories that try to calculate comprehension difficulty in sentence comprehension using complexity metrics (Lewis, 1993).

### 2.2.2.3 Frazier's Local Nonterminal Count

Frazier (1985) constructed a complexity metric based on the idea of Miller and Chomsky (1963) with the addition of a *local nonterminal count* that captures more fine-grained syntactic processing behavior than Miller and Chomsky's global nonterminal ratio. Frazier's local nonterminal count captures clusters of nonterminals of three adjacent terminals in the sentence by summing up the values of all nonterminal counts occurring in this window. Each terminal contributes a count of 1 to the local nonterminal count while the S node contributes 1.5 counts. The maximal local nonterminal count is the largest sum in a sentence. The higher the maximal local nonterminal count the more difficult a sentence is to parse. If two sentences have the same overall nonterminal-to-terminal node ratio the sentence with the higher maximal nonterminal count is more complex. In this way Frazier's metric can explain why (15) is easier to process than (14):

(14) That Ray smiled pleased Sue.  
3 2.5 1 1 1

(15) It pleased Sue that Ray smiled.  
2.5 1 1 1.5 2.5 1

The numbers below the words indicate the local nonterminal count for that particular word in the sentence. The maximal nonterminal count for (14) is 6.5 units for the first three words *that Ray smiled* and 5 units for (15) for the last three words *that Ray smiled*. Hence, Frazier's local nonterminal count makes the right predictions for these examples.

Frazier (1985) claims that her metric correctly accounts for the higher complexity of RCs in object position vs. subject position, extra-posed RCs vs. non-extra-posed counterparts and singly vs. doubly center-embedded RCs. Gibson (1991) argues that the metric fails to account for the fact that doubly-embedded relative



sentences are more complex than singly-embedded relative sentences:

(16) The man that the dogbit ate the cake that the  
 2.5 0 2.5 2.5 0 2 1 1 0 2.5 2.5  
 womansaw.  
 0 2

(17) #The man that the womanthat the dogbit saw  
 2.5 0 2.5 2.5 0 2.5 2.5 0 2 2  
 ate the cake.  
 1 1 0

Both sentences contain the same number of words and the same number of nonterminals and this results in the same local nonterminal count. The maximal local nonterminal count in both examples is 5 units which is even lower than in (14). According to Gibson (1991), Frazier's metric can only predict complexity or unacceptability correctly if the three words with the highest complexity are immediately adjacent, but this is not always the case. The fixed window of three words is not always suitable because nonterminals are sometimes more evenly distributed across the sentence even if the sentence causes parsing breakdown.

#### 2.2.2.4 Kimball's Principle of Two Sentences

Kimball (1973) proposed the *Principle of Two Sentences* which says "constituents of no more than two sentences can be parsed at one time" (Kimball, 1973, p. 33) to reflect the limits of short-term memory in the human processor. For instance, (18) is easy to understand while examples (19) and (20) incur processing difficulty at the most embedded word *Joe*. At this point there are three incomplete S nodes in the structure.

(18) That Joe left bothered Susan.

(19) # That that Joe left bothered Susan surprised Max.

(20) # That for Joe to leave bothers Susan surprised Max.

The principle of two sentences accounts for the contrast in acceptability between sentences with a sentential subject and sentences with embedded sentential subjects. Further, it correctly predicts the unacceptability of doubly center-embedded sentences because at the most embedded NP a third incomplete sentence node will be added to the structure generating an unacceptable sentence. However, Gibson (1991) points out that Kimball's principle would rule out a center-embedded RC inside a sentential subject even though the structure can be parsed. This is shown in example (21) (taken from (Gibson, 1991, p. 57)):

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

At the word *John* three S nodes are incomplete: the main clause, the subject sentence (*that the food tasted good*) and the embedded RC. Yet the sentence is acceptable. Additionally, Gibson (1991) shows that Kimball's principle incorrectly predicts processing difficulty for constructions with a RC inside a NP complement and constructions with a RC inside a pseudo-cleft.

## 2.3 Cognitive architectures

Two prominent types of cognitive architectures are discussed extensively in the psycholinguistic literature: *Memory-based* and *Expectation-based* Theories. Both make complementary predictions about processing difficulty in syntactic structures. Nevertheless, for both approaches there exists a vast amount of empirical evidence in psycholinguistics. The following sections will revisit several theories and their empirical support.

### 2.3.1 Memory-based Theories

Memory-based Theories argue that the human processor is constrained by limited memory resources. The section will start with a discussion of a selection of work on working memory: Gibson's

*Dependency Locality Theory*, followed by the *Retrieval-interference Theory* of Lewis and Vasishth, Rambow and Joshi's *Tenure*, and finally Marcus' *PARSIFAL* model. Other theories in the area of working memory architecture that are not discussed in detail are: the CAPS architecture and underlying CC Reader (Just and Carpenter, 1992), the PDP architecture (St. John and McClelland, 1990), the Unification Space model (Vosse and Kempen, 2000) and the NL-Soar comprehension model (Lewis, 1993).

### 2.3.1.1 Dependency Locality Theory

Gibson (1998) defines the Syntactic Prediction Locality Theory (SPLT) as a language-independent theory to explain distance effects in sentence comprehension. More recently, Gibson (2000) characterizes the Dependency Locality Theory (DLT) with a storage component that is not locality-based, in contrast to SPLT.<sup>3</sup> The key costs of syntactic operations for DLT are *storage cost* and *integration cost*. Storage cost is quantified by the number of incoming words or the number of predictions about incoming words that have to be stored in working memory. Integration cost is associated with connecting an incoming word to the syntactic representation of the sentence which includes retrieving the previous representation and its processed words from memory. Integration cost is sensitive to the distance between the dependent and its head (Gibson, 1998, 2000; Grodner and Gibson, 2005; Warren and Gibson, 2002). In particular, the locality cost (storage and integration cost) is a function of the number of new intervening discourse referents between the dependent and its head. Gibson (1998) defines a discourse referent as "an entity that has a spatio-temporal location so that it can later be referred to with an anaphoric expression, such as a pronoun for NPs, or tense on a verb for events" (Gibson, 1998, p.12). The more intervening discourse referents the higher the integration cost because the activation of a word decays over time. Both syntactic operations, integration and storage, use

---

<sup>3</sup>Locality means that syntactic predictions that are stored in memory for a longer time are more expensive.

the same pool of working memory resources (Gibson, 1998; Just and Carpenter, 1992). Gibson (1998) ascribes the reading times at a particular point in the sentence directly to the time an integration requires. The higher the memory load for storage, the fewer working memory resources are available, and the longer an integration takes. In this way the complexity of a sentence is defined as the maximal memory complexity during the parsing of a sentence as opposed to the average memory complexity.

One interesting point Gibson (1998) acknowledges is the fact that not only does the number of new discourse referents between the dependent and its head play a role for complexity calculations but also the syntactic complexity of the intermediate integrations of discourse referents. DLT ignores this intermediate structure. In a footnote, Gibson (1998, p.12) refers to the *Principle of Parsimony* of Crain and Steedman (1985) which states that in case of an ambiguity the reading with the fewest unsatisfied presuppositions, i.e. with the simpler discourse structure, is preferred. Gibson (1998) relates this principle to the construction of structures for new discourse referents. This thesis resumes this issue, namely, that a difference in the grammatical structure of intervening words makes a difference to the calculation of complexity of the sentence.

According to Gibson (1998), there are a number of phenomena in sentence comprehension that DLT can account for: (1) the higher complexity of object-extracted RCs compared to subject-extracted RCs; (2) the unacceptability of multiply center-embedded structures; (3) the higher complexity of center-embedded sentences compared to cross-serial dependencies; (4) the higher complexity of a heavy NP-shift to the end of a sentence compared to a heavy NP-shift to the beginning of a sentence; and (5) a considerable number of ambiguity effects. Interestingly, empirical support for the predictions of DLT are found not only in English (Gibson and Thomas, 1999; Grodner and Gibson, 2005; Warren and Gibson, 2002, 2005), but also in head-final languages like Chinese (Hsiao and Gibson, 2003; Gibson and Wu, 2013) and Japanese (Babyonyshev and Gibson, 1999).

The predictions of DLT for subject-extracted relative clauses (SRC) compared to object-extracted relative clauses (ORC) in English are outlined below. Later in this section DLT's predictions for multiply center-embedded sentences are discussed in more detail.<sup>4</sup>

DLT correctly predicts that SRCs are easier to understand than ORCs. The RCs in (22) and (23) modify the noun *reporter*. In (22) the word *who* marks the subject position of the verb *attacked* while in (23) *who* is in object position of *attacked*.

(22) The reporter who attacked the senator hoped for a story.

(23) The reporter who the senator attacked hoped for a story.

According to Gibson (1998) the source of complexity has nothing to do with lexical or plausibility differences since both structures contain the same words with the same plausibility relationships. The remaining source for processing difficulty is the amount of computational resources during parsing, i.e., storage cost (Gibson, 1998). At the verb *attacked* there are three open dependencies in the ORC (23): one between *the reporter* and its predicted verb; one between *the senator* and its predicted verb; and one between *who* and its predicted object position at the embedded verb. Further, this is the only point in the sentence when two integrations happen at the same time: the preceding subject and object are integrated into the structure. In the SRC example (22) there are at most two open dependencies; at the word *who* there is an open dependency between *who* and its predicted verb, and *the reporter* and its predicted verb.

### 2.3.1.2 Retrieval-interference Theory

Lewis and Vasishth (2005; (see also Vasishth and Lewis, 2006; Lewis et al., 2006)) present the *Retrieval-Interference Theory* of sentence processing which is implemented as part of the Adaptive Control of Thought-Rational (ACT-R) architecture (Anderson

---

<sup>4</sup>For a detailed discussion of further empirical support for SPLT see Gibson (1998).

et al., 2004). This approach simulates sentence processing as memory retrievals that are modulated by similarity-based interference and fluctuating activation. Retrieval is defined as restoring words from the previous context to integrate them into the current syntactic structure. The longer a word is stored the more its activation level decays over time, making the retrieval operation more difficult. Consequently the greater the distance between a dependent and its head the harder retrieval becomes, similar to the predictions of DLT. However, repeated retrieval can increase the activation level of a word stored in memory which leads to faster retrieval. In this way expectations about upcoming words can be sharpened and facilitate processing (see Vasishth and Lewis, 2006, for details from Hindi).

One aspect this approach takes into account is that incoming words can share syntactic features with words stored in memory, which is called *interference*. Interference can facilitate or hinder integration of a word into the syntactic structure (Van Dyke and Lewis, 2003; Vasishth and Lewis, 2006). A theory that assumes retrieval is harder with interference is *Similarity-Based Interference* (Lewis and Vasishth, 2005; Lewis et al., 2006; McElree et al., 2003).

For the SRC and ORC examples (22) and (23) the retrieval-interference theory predicts correctly that the ORC should cause processing difficulty on the embedded verb *attacked*. At this word the verb itself and the verb's argument NPs are retrieved to be integrated semantically with the verb. Since *reporter* and *senator* share several syntactic features (both are animate, singular and definite nouns) they interfere with one another which results in a longer retrieval process (Lewis and Vasishth, 2005; Vasishth and Lewis, 2006). In the SRC example only *reporter* is retrieved, causing less processing difficulty compared to the ORC sentence.

### 2.3.1.3 Tenure in an Embedded Pushdown Automaton

Rambow and Joshi (1994) propose a processing model and a complexity metric based on a grammar formalism, a Tree-Adjoining Grammar (TAG) (Joshi, 1990). In this way two superficially similar

sentences with distinct syntactic structures are predicted to induce a difference in processing complexity.

Originally, Joshi (1990) defined two complexity metrics: one that counts the maximum number of words that are stored in memory during the entire parsing of a sentence, and a second one that additionally takes into account for how long a word is stored in memory. Rambow and Joshi (1994) develop the second approach further and propose a metric that counts the words on the stack at each word in the sentence, and then sums up the scores for all words of the sentence, leading to a *cumulative score* for each word.

Rambow and Joshi evaluate their complexity metric using German data. In German, center-embedded structures are more difficult to understand than extra-posed structures which is correctly predicted by the complexity metric of Rambow and Joshi (1994). For extra-posed structures, the cumulative scores grow linearly with the number of embeddings, while for center-embedded structures the cumulative scores grow with the square of the number of embeddings.

More recently, Kobele et al. (2013) refined the complexity metric of Rambow and Joshi (1994). They introduced *Stack Tenure* which reflects the time a word is stored in memory (similar to Joshi, 1990). The approach assumes that a word that was stored earlier in the sentence is more costly than a word that was stored more recently, similar to the idea of working memory decay (Lewis and Vasishth, 2005) and long-distance dependencies (Gibson, 1998). Kobele et al. (2012) present empirical support from eye-tracking data for German cross-serial and Dutch nested verb clusters. Stack tenure correctly predicts that German readers show more difficulty, reflected in longer total reading times, with an increasing number of embeddings for cross-serial embeddings than Dutch readers show for the same number of nested embeddings.

### 2.3.1.4 Deterministic Parsing

Marcus (1980) proposed the *PARSIFAL model*. The model is based on the *Determinism Hypothesis* and operates strictly deterministically, i.e., the built syntactic structures are permanent and the model uses no backtracking. A sentence is parsed from left-to-right with an active node stack and a three-cell buffer lookahead<sup>5</sup> to model locally ambiguous sentences. The restriction of the lookahead is mandatory, otherwise the parser would not be deterministic. The buffer holds grammatical constituents of any type which range from single words up to a complete subordinate clause.

However, there are a couple of problematic examples for Marcus' parser. Jurafsky (1996) shows that the buffer of three constituents would incorrectly predict that sentence (24) would elicit a garden-path effect. The parser takes *my aunt* to be the direct object of *know* only to discover that this phrase is part of the NP *my aunt from Peoria*.

(24) I know my aunt from Peoria died.

Additionally, Marcus (1980) is not able build the correct interpretation for example (2) (repeated in (25) below) because the three-cell buffer cannot view *the horse* and the final word *fell* simultaneously. The structure for the initial NP will be built before it is moved out of the buffer, then the parser will build the main verb interpretation since there is no indication of a reduced RC until the final verb is parsed (Jurafsky, 1996).

(25) The horse raced yesterday fell.

An important issue missing from the parser is the preference or bias for one structural interpretation, as in the unrestricted race model. Lexical or structural preferences for an interpretation can guide or misguide the language processor in, for example, the case of a garden-path sentence.

---

<sup>5</sup>In a later chapter Marcus (1980) extends the buffer to five cells.



## 2.3.2 Expectation-based Theories

Expectation-based Theories argue that language processing is constrained by experience and/or generalization. If participants have more experience comprehending or producing certain syntactic structures, they will have less difficulty understanding and constructing these syntactic structures. In this section, *Word Order Frequency Theories* are firstly discussed using the example of Mitchell's *Linguistic Tuning Hypothesis* and Jurafsky's *Probabilistic Parser*. Then, Hale's *Surprisal* and *Entropy Reduction Hypothesis* are explained. Lastly, literature concerning the combination of Memory-based and Experienced-based Theories is outlined.

### 2.3.2.1 Word Order Frequency Theories

*Word Order Frequency Theories* predict that the higher the frequency of a certain word order in a language, the easier this word order is to process and comprehend (Bever, 1970; Crocker and Brants, 2000; Jurafsky, 1996; MacDonald and Christiansen, 2002; Trueswell et al., 1994).

One approach along these lines is the *Linguistic Tuning Hypothesis* of Mitchell et al. (1995). This hypothesis suggests that in the case of a structural ambiguity, the reader initially resolves the ambiguity on the basis of linguistic exposure or experience, i.e., the more frequently encountered structure is chosen. Hence, syntactic structures that are more frequent in the linguistic input influence parsing decisions immediately during online comprehension. Consequently, if only a less frequent structure is consistent with the analysis, a garden-path effect is predicted (Bever, 1970; Cuetos and Mitchell, 1988; MacDonald et al., 1994; Mitchell et al., 1995).

The tuning hypothesis can correctly predict RC attachment preferences in English and Spanish as shown in (26) and (27) (examples taken from Cuetos and Mitchell (1988, p. 77)):

- (26) El periodista entrevistó a [la hija del [coronel]] que tuvo el accidente.

- (27) The journalist interviewed [the daughter of [the colonel]]  
who had had the accident.

The RC *que tuvo el accidente/ who had had the accident* can either be attached to the NP *la hija/ the daughter* or to the NP *la hija del coronel/ the daughter of the colonel*. Spanish speakers prefer high attachment to the first NP while English speakers prefer low attachment to the second NP inside the prepositional phrase (Cuetos and Mitchell, 1988). Mitchell et al. (1995) attribute the difference in attachment preferences to the difference in frequencies of the distribution of similar ambiguities in these languages. In Spanish, adjectives follow the noun, whereas in English they precede it (Cuetos and Mitchell, 1988).

A second approach was put forward by Jurafsky (1996). His model estimates top-down (grammatical preferences) and bottom-up (lexical preferences) information simultaneously in a parallel ranked parser. The parser ranks alternative structural analyses according to their conditional probability. A pruning function eliminates low-ranked interpretations while high-ranked structures are maintained. The parser can account for a variety of syntactic garden-path effects (Jurafsky, 1996).

To model the example:

- (28) # The horse raced past the barn fell.

The probabilistic parser assigns a probability value of .92 to the main-clause interpretation of *race* and .08 to the reduced RC reading since the main-clause structure is preferred. Additionally, the reduced RC reading requires the NP *the horse* to be the object of *race* with an additional rule [.14] NP → NP XP. Taken together, these two requirements make the main-clause interpretation 82 times more probable than the reduced RC reading and lead the parser into a garden-path (Jurafsky, 1996).

### 2.3.2.2 Surprisal

Hale (2001) proposed *surprisal* which calculates fine-grained expectations of upcoming words based on the *prefix probability*. Prefix probability is the sum of all probabilities of previous input words. Surprisal is the negative logarithm of the prefix probability, in other words, surprisal accounts for the probability of syntactic structures that are eliminated at the current word (Boston et al., 2008; Hale, 2001).

The word-by-word probability values are calculated using a probabilistic Early parser generating multiple predictions on the basis of a probabilistic context-free phrase structure grammar (PCFG) (Hale, 2001). Each rule in the grammar has a probability estimated by frequencies of occurrences in a corpus (e.g., Penn Treebank). The product of the probability of each grammar rule used to build the syntactic structure of the sentence represents the probability of this sentence in the language. Surprisal accounts for changes in conditional probability from one word to the next during incremental parsing. If a word is unexpected in a particular context its surprisal value will be high reflected in a high processing difficulty.

Surprisal captures a variety of complexity effects, for instance the higher complexity of object compared to subject RCs, speed-up effects at the final verb, word order asymmetries in German and several garden-path effects (Boston et al., 2008, 2011; Demberg and Keller, 2008b; Hale, 2001; Levy, 2008).

Boston et al. (2008) tested the predictability of surprisal for eye movement data in German using two different grammar types: a hierarchical phrase-structure grammar (PCFG) and a word-to-word dependency grammar. PCFG uses unlexicalized surprisal while the dependency grammar uses fully lexicalized surprisal (Roark et al., 2009). Lexicalized surprisal takes into account how much the exact words contribute to the calculation of structural and lexical probabilities. Unlexicalized surprisal uses only structural probabilities and does not include word frequencies or the probability of a specific part-of-speech tag being assigned to a

word. The results from Boston et al. (2008) show that surprisal can predict reading times and regressive eye movements independent of word length and frequency for both grammar types. According to them the predictions cover so-called first-pass dependent measures of eye movement (e.g., single fixation duration, first fixation duration, regression probability) as well as non-first-pass dependent measures of eye movement (e.g., regression path duration, total reading time).

Demberg and Keller (2008b) investigated the difference between lexicalized and unlexicalized surprisal using the Dundee Corpus, an eye-tracking corpus based on English newspaper texts. The results corroborate the findings by Boston et al. (2008) even though Boston and colleagues used a German corpus with manually constructed sentences. The results show that unlexicalized surprisal is a significant positive predictor of processing complexity, i.e., the higher the value of unlexicalized surprisal the longer the reading times on a word, while lexicalized surprisal failed to show such an effect. Demberg and Keller (2008b) conclude that unlexicalized (or structural) surprisal instead of lexicalized surprisal can explain processing complexity.

### 2.3.2.3 Entropy Reduction

The theory of *Entropy Reduction* (Hale, 2003a, 2006) calculates the probability of multiple possible structures at each word in the sentence. The structure with the highest probability, or lowest entropy, is chosen. In other words entropy calculates how much information a word conveys regarding the rest of the sentence. At the beginning of a sentence the uncertainty about the next word is the conditional entropy. As the next word is parsed, entropy is reduced. The amount of reduction reflects the number of alternative structures that had to be discarded at this point. If the amount of reduction is high and there is a large drop in entropy then the theory predicts processing difficulty.

Hale (2003a) calculates the entropy reduction word-by-word on the basis of a probabilistic context-free grammar. To obtain the

entropy value for a complete sentence, the intermediate entropy values are summed up, resulting in the total processing difficulty of reading a particular sentence (Hale, 2006).

Hale (2006) and Hale (2003b) use entropy reduction with minimalist grammars that are mildly context-sensitive. The probabilities of the grammar rules were calculated based on the Brown corpus. Entropy reduction calculated on the basis of a minimalist grammar correctly predicts the asymmetry between subject-object RCs.

Originally, Hale (2006) proposed that entropy reduction is lexicalized. Roark et al. (2009) extended this definition and defined a syntactic version of entropy by calculating the entropy up to the pre-terminal node only. Roark and colleagues' results show only an effect of syntactic entropy and not lexical entropy. However, they argue that this finding could be due to sparse data in the Brown corpus on which they trained their parser.

### **2.3.3 Memory-based and Experience-based Theories Combined**

Since memory-based and experience-based theories make opposite predictions in sentence comprehension, researchers have tried to find support for a combination of both approaches to obtain a complete processing theory (Boston et al., 2008, 2011; Demberg and Keller, 2008b; Gibson and Wu, 2013; Levy, 2008; Levy and Keller, 2013; Levy et al., 2013; Vasishth and Drenhaus, 2011).

For instance, Demberg and Keller (2008b) directly compared predictions of the memory-based account of DLT (Gibson, 1998, 2000) and the experience-based theory of surprisal (Hale, 2001) against the Dundee Corpus. Both theories are orthogonal to one another and make complementary predictions about the source of processing difficulty. The integration component of DLT is a backward-looking measure capturing the memory cost when a head has to be integrated with its preceding dependents. Conversely, surprisal is considered to be a forward-looking measure predicting the cost of a word when this word occurs in

an unexpected context. Another important difference between the two approaches is that DLT only assigns values to nouns and verbs, which limits the power of this approach,<sup>6</sup> while surprisal is calculated for all words in the sentence. Demberg and Keller (2008b) suggest that a complete theory of sentence processing needs to combine both approaches: DLT and surprisal. According to them, an upcoming word has two costs: the cost of integration into the built syntactic structure and the cost of discarding alternative structures that become implausible with the integration of the new word.

Boston et al. (2011) also put forward the idea of a combination of memory-based and expectation-based theories. Predictions of surprisal (Hale, 2001) and cue-based retrieval (Lewis and Vasishth, 2005) are compared directly using the same incremental parsing algorithm (Nivre, 2004) trained on a German corpus of newspaper text and tested on German eye-tracking data (Potsdam Sentence Corpus). Again, these two approaches make different predictions about the source of processing difficulty.

The results of Boston et al. (2011) show that surprisal, but not retrieval, could predict processing difficulty in fixation durations and regression probability at a low beam-width. With an increase of the beam-width, i.e., an increase of parallelism in the model, surprisal continued to be a good predictor of the data for all measures, and retrieval caught up. At a maximum beam-width of  $k = 100$  the combination of surprisal and retrieval yielded the best fit to the eye-tracking data for all measures except regression probability; here, the best model was the one with only surprisal as a predictor.

Gibson and Wu (2013) tested the predictions of both approaches empirically in the context of subject-extracted and object-extracted RCs in Chinese. According to Gibson and Wu (2013), working-memory based accounts predict that ORC should be easier to process than SRC, whereas experience-based theories predict the op-

---

<sup>6</sup>Warren and Gibson (2002) suggest an integration calculation that also takes into account the givenness status of a noun, e.g., for pronouns, but they leave the implementation to future work.

posite because SRC structures are more frequent in Chinese than ORC structures. The experimental results support the prediction of working-memory based accounts. ORC sentences are read faster than SRC sentences in a context licensing both structures. Further, Gibson and Wu (2013) show that for experience-based theories the processing difficulty of ORC as opposed to SRC sentences depends on the types of NPs in the RC and the head noun modifying the RC. Gibson and Wu (2013) cite Traxler et al. (2002) who showed that ORCs like *the rock that the boy threw* in which an inanimate patient is modified by an animate subject are easier to process than ORCs like *the mountaineer that the boulder hit* in which an animate patient is modified by an inanimate subject. These findings are compatible with predictions from experience-based theories but not with memory-based accounts. Hence, they find evidence for both approaches.

Levy et al. (2013) present evidence supporting both memory-based and expectation-based theories about subject-extracted and object-extracted RCs in Russian. They compare the predictions of DLT (Gibson, 1998, 2000) and cue-based retrieval (Lewis and Vasishth, 2005) as memory-based theories with the predictions of surprisal (Hale, 2001) and entropy reduction (Hale, 2003a, 2006) as experience-based approaches. The results show a monotonic increase of reading times on the embedded verb as more words intervene between the relative pronoun and the embedded verb. This finding is in line with predictions of both memory-based theories (DLT and cue-based retrieval) but not with the experience-based theories (surprisal and entropy reduction). Levy et al. (2013) found another interesting effect that is predicted by experience-based rather than memory-based theories: an increase of reading time at the RC-initial accusative object NP where this NP is unexpected.

The results corroborate the findings of Staub (2010) in English SRC vs. ORC sentences whose results show similar effects: (1) an increase of reading time on the embedded verb and (2) an increase

in regressions out of the initial determiner *the* of *the fireman* in (30) compared to (29).<sup>7</sup>

- (29) The employees hoped that the fireman noticed the people who were still in the building.
- (30) The employees that the fireman noticed hurried across the open field.

Interestingly, Staub (2010) states that processing difficulty at the embedded verb and the subject of the ORC sentence is predicted by the cue-based retrieval model of Lewis and Vasishth (2005). The former is caused by retrieval of the verb from memory, and the latter by “the need to revise an expectation for an SRC when the ORC determiner is encountered” (Staub, 2010, p. 82). Vasishth and Drenhaus (2011, p. 75) suggest that expectation and memory resources interact with each other: “expectation plays a dominant role only when working memory load is relatively low”.

---

<sup>7</sup>Both sentences contain the same lexical material in a parallel linear order so that the eye movement data in the critical region is comparable. That is why an ORC sentence is compared to a verbal complement clause instead of an SRC sentence.



### 3 The Nature of the Human Parser

Previous research on working memory quantifies the distance between the dependent and the head in terms of *linguistic units* such as discourse referents (Gibson, 1998, 2000; Warren and Gibson, 2002), in terms of *time*, i.e., activation levels of incoming material (Lewis and Vasishth, 2005; Lewis et al., 2006) and in terms of *hierarchical syntactic structure*, i.e., number of intervening phrase-structure nodes (Frazier, 1985; O'Grady, 1997; Hawkins, 1999). The two broad classes of theories that we will distinguish here are theories of *linear distance* that attribute the greater difficulty at the head to the greater number of intervening words (Gibson, 2000; Lewis and Vasishth, 2005) and theories of *structural distance* that explain processing difficulty as the higher number of intervening phrase-structure nodes in the syntactic tree (Hawkins, 1994; O'Grady, 1997).

Since this thesis focuses on structural-based complexity, this chapter will mainly review previous work on structural-based approaches and from time to time compare their predictions to predictions of linear-based theories. Firstly, Section 3.1 outlines the *Derivational Theory of Complexity* that was the first theory proposing a relationship between grammatical operations in a sentence and processing complexity. The second part of this chapter summarizes work on the *Structural Search of the Parser* using at first filler-gap dependencies as an interesting phenomenon to distinguish linear and structural distance. In filler-gap dependencies the filler and the gap can be separated by a number of words in a sentence but the parser is still able to link the filler and the gap. Numerous studies in different languages, e.g. English, German, Japanese and Korean, show that the processing difficulty associated with different syntactic structures can be

accounted for by structural distance rather than linear distance. A second example for structural search is the licensing of negative polarity items by a c-commanding negator. Thirdly, Section 3.3 *Structurally-mediated Reactivation* outlines studies on the resolution of pronouns and reflexives. Empirical studies on the resolution of pronouns and reflexives suggest a structurally-mediated reactivation of the potential antecedent during online sentence processing. Finally, *Computational Models* in the previous literature that implement structural-based approaches by using grammatical knowledge will be reviewed.

### **3.1 Derivational Theory of Complexity**

Quite early in the literature, the Derivational Theory of Complexity (DTC) was proposed by Fodor and Garrett (1966). This theory constitutes a direct relationship between the complexity of a sentence and the grammatical operations employed on the syntactic structure. The more grammatical operations are necessary to parse the syntactic structure the higher the perceptual complexity (Fodor and Garrett, 1967).

Fodor and Garrett (1966) cite a study by Savin and Perchonock (1965) that investigated the memory requirements of various sentence types with regard to different transformations (e.g., negation, passivization, etc.). The hypothesis of Savin and Perchonock states that the complexity of a sentence is constituted by the number of transformational rules applied to the syntactic structure of the sentence as opposed to the number of words in the sentence. Participants were asked to recall sentences or a set of unrelated words. Savin and Perchonock (1965) assumed that the more words are successfully recalled (provided that the sentence was recalled correctly as well) the less costly it is to memorize the sentence structure, presuming that the distinct sentence structures are encoded differently in memory and therefore occupy a specific amount of space. The results of Savin and Perchonock (1965) confirm this hypothesis such that more words were recalled for simple

sentence types, e.g. active sentences, *wh*-sentences, questions, as opposed to more complex sentence types, e.g. negated questions, passivized questions, negative passive sentences. They conclude that the length of the sentence in terms of words does not account for these results. However, Fodor and Garrett (1966) criticize the fact that the materials were not controlled for length and meaning.

Fodor and Garrett (1967) suggest that the complexity of a sentence is not only a function of the number of applied grammatical operations to the syntactic structure, but also the available parsing cues in the sentence that might facilitate parsing operations. The examples in (31) and (32) (taken from Fodor and Garrett, 1967) differ only in the word *whom* which is present in (32).

(31) The man the dog bit died.

(32) The man *whom* the dog bit died.

The relative pronoun makes the second example less complex than the first one because the pronoun provides a cue for the relationship between the two noun phrases (NPs) in the sentence. *The man* is the object and *the dog* is the subject in the embedded clause in both sentences. In (31) the subject/object relationship becomes clear only at the verb because of the verb semantics. Fodor and Garrett (1967) conclude that the pronoun provides a good cue about the syntactic structure of the sentence even though it increases the distance for the dependencies in the structure.

## 3.2 Structural Search of the Parser

This section summarizes work related to the structural search of the parser. The phenomena in this section show that the human parser is sensitive to structural constraints. Further, the section highlights examples from the literature that try to distinguish if linear-based or structural-based distance between a dependent and its head can explain processing difficulty. In this section we will argue in favor of structural-based distance

accounting for processing difficulty associated with various syntactic structures in different languages, e.g. English, German and Korean. Section 3.2.1 reviews work on filler-gap dependencies and Section 3.2.2 gives an overview on the licensing of negative polarity items by a c-commanding negator.

### 3.2.1 Filler-Gap Dependencies

Filler-gap dependencies are an interesting phenomenon indicating that the parser implements a *structural search* for dislocated elements in the sentence. Behavioral data suggests that the filler, e.g. a displaced wh-element like *who*, is stored in working memory until it is reactivated at the original position of the moved element, i.e., its gap (Clifton and Frazier, 1989; Nicol and Swinney, 1989; Fiebach et al., 2002), because the filler and the gap are dependent on each other (cf. Fodor, 1989). The longer it takes to memorize the filler the higher the effort to find the gap, resolve the dependency and release memory demands (Hawkins, 1999). Even though a number of words can be placed between the filler and its gap to create a long-distance dependency, the processor is still able to correctly link the two (Wagers, 2013).

Quite early, theories of filler-gap dependencies (Wanner and Maratsos, 1978) suggest that the filler is stored in a special memory, the HOLD cell, and is retrieved from there to be assigned to its gap. The HOLD cell functions as a store for an item whose grammatical function cannot be assigned in the current context, e.g. the head of an NP to which a relative pronoun (e.g., *who*) refers in a relative clause (RC). HOLD is part of a lexical-expectation model, the augmented transition network (ATN), proposed by Wanner and Maratsos (1978). The ATN consists of two main components: a processor and a transition network grammar. Additionally a perceptual and a semantic analyzer guide the parsing. All these parts use the same lexicon and a common working memory. The ATN identifies word categories, phrase boundaries and grammatical functions, and stores them in the working memory. Regarding filler-gap dependencies the model

is based on two assumptions. First, a gap is simply an empty NP. Second, the parser ranks options for the verb subcategorization frame for a lexical item. If a verb is usually transitive, e.g. read, the parser will predict a direct object in the syntactic structure. If, however, there is no overt direct object then the parser posits a gap at this position. The ATN is one of the first psychological models of syntactic processing which assumes a syntactic structure for working memory.

In the previous literature two opposing views to explain parsing behavior for filler-gap dependencies have been discussed. Either the parser based on a grammar formalism would be driven by the syntactic features of the head and project these onto the phrase, i.e., *head-driven parsing* (Bresnan, 1982; Frazier, 1987; Pritchett, 1991) or the grammar rules predict upcoming phrases immediately and the parser does not wait for the head of the phrase, i.e., *incremental parsing* (e.g., Clifton and Frazier, 1989; Bader and Lasser, 1994).

The projection principle of the Government and Binding Theory assumes the former, namely that the parser is guided by features of the head of a structural phrase and therefore tries to identify the head of a phrase first. Hence, speakers of head-final languages, e.g. Japanese, should follow different parsing principles from speakers of head-initial languages such as English. According to the theory, users of head-final languages would delay parsing decisions until they encounter the head of the phrase. This hypothesis led to the development of generative grammars (Chomsky, 1981, 1995) and head-driven parsers (Pritchett, 1991). Generative grammars project the syntactic structures from the lexicon without the application of phrase-structure rules. The lexicon consists of rich lexical entries which interact, for instance, with binding principles to build a grammatically well-formed syntactic structure. Pritchett (1991) was one of the first who introduced head-driven parsing. He proposed that the syntactic features of the head, e.g. word category, arguments, cannot be projected before the head is encountered and so are left unattached until then.

In the model, licensing principles specify attachment, e.g. theta-attachment (Bader and Lasser, 1994).

The alternative view to head-driven parsing comes from incremental parsing models which assume that the parser uses predictions of grammar rules (or lexical information) such that a determiner would posit an NP directly without waiting for the noun as the head of the phrase, or a verb would be predicted solely on its argument structure (Clifton and Frazier, 1989; Bader and Lasser, 1994, among others). Hereby in head-final languages a partial phrasal node is projected and incoming arguments are connected incrementally before the verb as the head of the verb phrase (VP) arrives.

Filler-gap dependencies are an interesting test case to disentangle these two assumptions about the human parser. For instance, a possible gap in position of the direct object of a verb might be posited as soon as the VP is predicted, or later during parsing as the verb at the head of the phrase is encountered and the verbal features are determined as either transitive or ditransitive. Empirical evidence indicates no delay of parsing decisions until the head of a phrase is encountered (Clifton and Frazier, 1989).

For instance, in Dutch, heads of VPs are generally head-final. Frazier (1987) directly tested if the syntactic analysis of a filler-gap dependency is delayed until the head of the sentence-final VP arrives. She used subject and object RCs in which the verb cannot disambiguate the grammatical function of the NPs in the RCs because these sentences are verb final in Dutch. If the head-driven parsing approach is correct, then there would be no differences in reading times between subject and object RCs because the parser delays the processing of the syntactic structure until the verb is read in both, subject and object RCs. If, however, the parser works incrementally and the Active Filler Hypothesis is correct, then the parser will prefer subject RCs over object RCs because Dutch readers immediately postulate a gap when encountering the relative pronoun *die* (who). The relationship between the filler and the

gap is resolved earlier in a subject RC than in an object RC, and before the final verb is read. Frazier conducted a self-paced reading experiment with ambiguous and non-ambiguous subject and object RCs. After reading the sentences participants had to answer comprehension questions asking directly for the interpretation of the head of the relative clause. The results of Frazier's self-paced reading experiment show that subject RCs were preferred over object RCs indicating that participants preferably identified the head of the RC as the subject. Numerically the final frame of the object RCs was read more slowly compared to the same frame in the subject RCs, however this difference was not significant. These results are not in line with predictions of the head-driven approaches that would predict no differences between subject and object RCs.

More data against the head-driven parsing approach comes from an experiment in German by Bader and Lasser (1994). Similar to Dutch, German is head-final for VPs in embedded sentences. Bader and Lasser tested if all attachments are delayed until the end of the sentence when the final verb is read. They conducted a self-paced reading experiment with ambiguous and non-ambiguous active and passive sentences. The ambiguity involved a phrase-structure ambiguity for one of the NPs in the embedded sentence. This NP had two possible attachment sites: as a direct object under a VP headed by V1, and as a subject under a VP headed by V2. The disambiguating element was the auxiliary V2 at the end of the sentence which signalled either a passive or an active reading of the sentence. Head-driven parsing predicts that the parser postpones phrase-structure decisions until the end of the sentence when the final verb is read. However, incremental parsing approaches predict that the parser prefers one structure over the other and has to revise this preference when the final verb disambiguates towards the dispreferred interpretation. In this case, the results would show processing difficulty for one structure but not the other. Bader and Lasser's results show that the human parser predicts the node of a VP in an embedded sentence and attaches verbal arguments to it before the final verb

occurs. In particular, participants show a slow-down at the auxiliary V2 of ambiguous sentences when the auxiliary signalled that the NP was the direct object of the embedded verb V1. These results show that the parser builds the syntactic structure incrementally and does not wait until the head of the phrase (V for VP) is encountered.

Evidence in English comes from Clifton et al. (1991) who investigated PP-attachment ambiguities. They found that PPs are immediately attached to the VP regardless of specific lexical information from the noun or verb preceding the PP. To summarize, these studies suggest that the human processor works incrementally and without waiting for relevant information about the head of the phrase to arrive.

Another interesting question in the domain of the human parser is: are working memory and syntactic integration processes distinct or unified processes? Evidence for the division of working memory and integration processes regarding the processing of filler-gap dependencies comes from results in event-related brain potential (ERP) experiments (King and Kutas, 1995; Fiebach et al., 2002; Hestvik et al., 2012; Ueno and Garnsey, 2008). For German, Fiebach et al. (2002) observed a sustained left-anterior negativity (LAN) for filler-gap dependencies at the position of the gap. Additionally, the amplitude of the negativity increased with increasing distance from the filler and the negativity was modulated by individual differences in working memory capacity, with a stronger and more distributed effect for participants with low working memory capacity. Fiebach and colleagues interpret this effect as reflecting working memory processes for keeping track of the dislocated object. Syntactic integration of the filler at the gap position was reflected by a positive-going ERP effect between 400 and 700ms, independent of the distance. This effect did not interact with working memory capacity. Hence, Fiebach et al. (2002) conclude that the local integration process is independent from working memory resources. Hestvik et al. (2012) found similar results for English filler-gap dependencies. Participants exhibited



a LAN in response to a filled gap NP when participants tried to link the filler to the gap which was already filled. Following the LAN, Hestvik and colleagues observed a P600 when participants tried to integrate the NP into the current syntactic structure where it cannot be integrated because the position is occupied by the postulated gap. To summarize, these findings from ERP experiments can differentiate between processes related to working memory and integration of syntactic structure. However, the question remains if the sustained negativity is driven by the number of words stored in memory or the structural complexity of the intervening material.

Data from Korean filler-gap dependencies (Kwon et al., 2010) might shed more light on this question. Kwon et al. (2010) investigated explicitly the distinction between predictions for linear distance and structural distance of subject-extracted (SR) and object-extracted (OR) relative clauses in Korean. Linear distance describes the number of words between the filler and the gap (e.g., Gibson, 1998; Warren and Gibson, 2002). Structural distance is characterized in terms of hierarchical syntactic structure (e.g., Chomsky, 1981); the object gap position is more deeply embedded in the syntactic structure compared to the subject gap position (O'Grady, 1997), hence there are more syntactic nodes intervening between the filler and the gap in ORs compared to SRs.

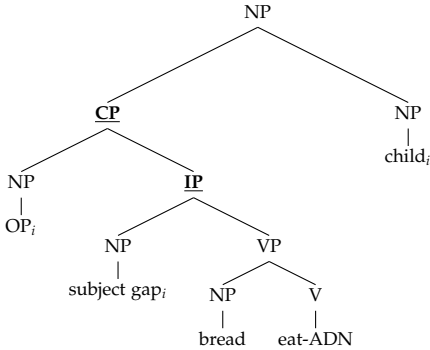
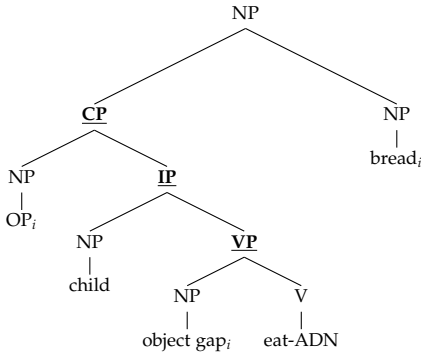
In Korean the gap is encountered before the filler, which is the head noun (pre-nominal RC), in contrast to post-nominal RCs in English. Additionally, there is no relative pronoun that initiates the RC or signals the gap, while English has relative pronouns, e.g. *who* and *that*. Instead, an adnominal marker *-n* suffixed to the sentence-final verb indicates the RC structure. This said, a gap is postulated at the sentence-initial NP-ACC in SRs (*senator-ACC* in (33)) and at the embedded predicate in ORs (*attack-ADN* in (34)) since the transitive verb *attack* will signal a missing argument. Hence, the linear distance between the gap and the filler is longer for SRs such as (33) compared to ORs like (34). Reactivation theories suggest that all parsed words are reactivated at the embedded

verb *attack* in order to be integrated into the structure of the RC. The verb position is the same in both sentences. Therefore the linear distance to the filler (*journalist*) is the same and reactivation theories would not predict any processing differences between SRs and ORs in Korean (Kwon et al., 2010).

- (33) [ \_\_\_<sub>i</sub> uywon-ul kongkyekha-n] enlonin<sub>i</sub>-i  
 [ \_\_\_ senator-ACC attack-ADN] journalist-NOM  
 yumyengha-ta  
 is.famous-DECL  
*The journalist who attacked the senator is famous.*

- (34) [uywon-i \_\_\_<sub>i</sub> kongkyekha-n] enlonin<sub>i</sub>-i  
 [senator-NOM \_\_\_ attack-ADN] journalist-NOM  
 yumyengha-ta  
 is.famous-DECL  
*The journalist who the senator attacked is famous.*

A structural-based distance proposed by O'Grady (1997) predicts increased processing difficulty with an increased number of intervening XP categories (e.g., IP, VP, NP, etc. (O'Grady, 1997, p.136)) (for a similar approach see Hawkins, 1994). This approach proposes less processing difficulty for SRs compared to ORs, because the subject gap is closer to the head noun in terms of phrasal nodes in the SRs than the object gap is to the head noun in ORs. This asymmetry in processing should be observed at the head noun where the filler is associated with the gap. Figure 3.1 illustrates these predictions of the structural-based account.

(a) SR: 2 XPs, *the child who ate bread*(b) OR: 3 XPs, *the bread that the child ate***Figure 3.1:** Phrase structure trees of SRs and ORs, Kwon et al. (2010, p. 7)

Kwon et al. (2010) tested three different syntactic structures both as an SR and as an OR: (35) subject-modifying RCs, (36) scrambled object-modifying RCs and (37) in-situ object-modifying RCs. The critical regions are the embedded verb *invited* and the following head noun *conductor* and *senator*.

(35) Subject-modifying RCs

[ yumyenghan sengacka-lul/ka chwukcen-ey chotayha-n]  
[ famous vocalist-ACC/NOM festival-to invited-ADN]  
cihwuyca-ka uywon-ul kongkongyehi  
conductor-NOM senator-ACC publicly  
moyokhay-ss-ta  
insult-PST-DECL

SR: The conductor who invited the famous vocalist to the festival publicly insulted the senator.

OR: The conductor who the famous vocalist invited to the festival publicly insulted the senator.

(36) Scrambled object-modifying RCs

[ yumyenghan sengacka-lul/ka chwukcen-ey chotayha-n]  
[ famous vocalist-ACC/NOM festival-to invited-ADN]  
cihwuyca-lul uywon-i kongkongyehi  
conductor-ACC senator-NOM publicly  
moyokhay-ss-ta  
insult-PST-DECL

SR: The senator publicly insulted the conductor who invited the famous vocalist to the festival.

OR: The senator publicly insulted the conductor who the famous vocalist invited to the festival.

(37) In-situ object-modifying RCs

[ yumyenghan cihwuyca-ka sengacka-lul/ka  
[ famous conductor-NOM vocalist-ACC/NOM  
chwukcen-ey chotayha-n] uywon-ul kongkongyehi  
festival-to invited-ADN] senator-ACC publicly  
moyokhay-ss-ta  
insult-PST-DECL

SR: The famous conductor publicly insulted the senator who invited the vocalist to the festival.

OR: The famous conductor publicly insulted the senator who the vocalist invited to the festival.

The dependent eye-tracking measures were gaze duration, regression-path duration and rereading time. Kwon et al. (2010) associate later stages of processing with regression-path duration and rereading time and expect processes related to memory retrieval to show up in these measures.<sup>1</sup> Overall ORs exhibit longer reading times at the head noun (filler) than SRs, most prominently for subject-modifying RCs; this was less marked for scrambled object-modifying RCs, and there was no effect for in-situ object-modifying RCs. Linear distance accounts predict the opposite finding, i.e., a disadvantage of SRs compared to ORs or no processing advantage for either one. Therefore, linear distance cannot explain these results. Structural-based accounts can account for these results and predict longer reading times for ORs compared to SRs because of the longer structural distance between the gap and its filler in the ORs (Kwon, 2008).<sup>2</sup>

Kwon et al. (2010) suggest that the absence of an effect for the in-situ object-modifying RCs might be due to a structural ambiguity in the materials. In particular, the in-situ object-modifying RCs in the SR condition contain successive NPs with the same case marking whereby the second NP can be initially parsed as the object of the main clause instead of the embedded clause. This ambiguity is resolved only at the head noun position, introducing a confound in the materials. There is no such ambiguity in the OR condition. Additionally, the absence of an effect for the linear-based accounts might be due to an inherent structural ambiguity of RCs in Korean (Kwon et al., 2010). Korean is a *pro*-drop language in which arguments can be dropped and the parser might interpret the gap position in the RC as a *pro* rather than a gap. This ambiguity is again resolved at the head noun which is obligatory only for RCs but not for *pro*.

---

<sup>1</sup>Categorizing eye-tracking measures into early and late measures of processing is quite controversial in the literature. Still, we will describe the claims of Kwon et al. (2010) using these categories.

<sup>2</sup>Furthermore, the advantage of SRs over ORs is also in line with predictions of experience-based models, since SRs are more frequent than ORs in Korean.

To rule out influences of these two structural ambiguities, Kwon et al. (2010) added context sentences promoting the RC reading to their second experiment and tested SRs and ORs with and without preceding context. If linear distance between the gap and its filler in combination with structural ambiguity can explain the results of experiment 1, then an interaction of gap type and context is predicted by these approaches. In particular, when there is no context then there is a disadvantage for ORs compared to SRs reflected in longer reading times at the head noun where the filler is integrated into the syntactic structure. On the other hand, if structural distance can account for the results of experiment 1 then ORs should take longer to read compared to SRs at the head noun regardless of the use of context (no interaction is predicted).

The findings of experiment 2 by Kwon et al. (2010) are in line with the predictions of structural-based accounts. ORs show more processing difficulty, i.e., longer reading times, than SRs without any influence of context which is also consistent with results from experiment 1. Even though rereading times show an interaction of gap type and head noun type (subject, in-situ object, and scrambled object head nouns) at the embedded verb, this effect is different from the predictions of linear-based accounts, since ORs were read slower than SRs at this point. The interaction results from a smaller asymmetry between ORs and SRs when there was a preceding context as opposed to a bigger asymmetry without context. Therefore these results are consistent with the prediction of structural-based accounts.

A question that could not be answered by Kwon et al's results is whether structural distance is characterized by the number of intervening phrase-structure nodes as O'Grady (1997) proposed, or by the structurally more deeply embedded object gap compared to the less deeply embedded subject gap. This way complexity would be defined by structural position of the gap in the syntactic structure rather than the distance between the gap and its filler.

Similar results for Japanese RCs were observed by Ueno and Garnsey (2008). As in Korean, RCs are also post-nominal in

Japanese and no overt relativizer signals the RC. Ueno and Garnsey (2008) conducted a self-paced reading and an ERP experiment to investigate if processing difficulty related to filler-gap dependencies in Japanese RCs can be explained by linear or structural distance. Interestingly, English and Japanese RCs have similar syntactic structures. In particular, the object gap position in ORs is more deeply embedded than the subject gap position in SRs for both languages (similar to Korean in Figure 3.1). Therefore, despite different surface structures resulting in longer linear distance for ORs in English and longer linear distance for SRs in Japanese, both languages show longer structural distances for ORs compared to SRs (Ueno and Garnsey, 2008). Hence, linear and structural distance yield different predictions for Japanese ORs. Previous reading time studies have shown longer reading times for ORs compared with SRs in Japanese and Korean (Miyamoto and Nakamura, 2003; Nakamura and Miyamoto, 2006; Kwon, 2008; Kwon et al., 2010) which suggests that structural distance accounts for these results rather than linear distance.

Ueno and Garnsey (2008) replicated these results in their self-paced reading experiment. They found elevated reading times for the ORs compared to the SRs at the head noun, which supports the structural account for the integration process of the filler-gap dependency. As a second experiment Ueno and Garnsey (2008) conducted an ERP experiment and found similar results to Fiebach et al. (2002). In particular, Ueno and Garnsey observed an anterior negativity with a typical latency and scalp distribution similar to the LAN, but it was bilaterally distributed instead of lateralized. The component is linked to the time window of the RC region (SR or OR) and is interpreted to reflect working memory demands associated with the filler-gap dependency. Upon seeing the transitive verb in ORs, the parser postulates the gap and stores it in working memory until it is filled by the head noun following the RC. Postulating the gap in SRs seems easier and leads to no observable negativity in the ERP results (Ueno and Garnsey, 2008). An alternative interpretation given by Ueno and Garnsey would

be that the gap in object position in ORs is more deeply embedded than the subject gap in SRs and induces more working memory load, which would support the structural distance interpretation.

The results in the post-RC region exhibit a component with a scalp distribution like the P600, but it is long-lasting as opposed to a local peak which is usually observed for a P600 effect. Ueno and Garnsey discuss alternative interpretations of this finding, however they conclude that the observed component is a P600 reflecting the integration process. This interpretation is consistent with previous results (Fiebach et al., 2002; Kaan et al., 2000) which suggest that the P600 reflects integration difficulty proportional to the amount of energy used to reactivate previous predictions and integrate these into the current syntactic structure. Hence, the P600 seems to reflect the higher integration cost associated with a more deeply embedded object gap in the ORs as opposed to the subject gap in SRs. The greater embedding requires more resources for retrieval of the object gap because of greater structural distance to its filler (head noun) (Ueno and Garnsey, 2008). To summarize, the findings of Ueno and Garnsey (2008) suggest that structural distance rather than linear distance accounts for the higher processing difficulty associated with ORs compared to SRs in Japanese.

### 3.2.2 Licensing of Negative Polarity Items

Another interesting phenomenon which provides insights into the processing and mental representations of structural relations during sentence comprehension is the *licensing of negative polarity items* (NPIs). NPIs such as *ever* are only licensed when they appear in the context of a negator like *no*. Additionally, *no* must c-command an NPI such as *ever*.<sup>3</sup> In other words, the NPI

---

<sup>3</sup>In this thesis we will focus on the negative contexts of NPIs by using examples with *ever*. However, there are other licensing contexts for NPIs such as yes-no questions, wh-questions, S-conditionals etc. (Vasishth et al., 2008). Additionally, there are even contexts in which the negation does not overtly c-command the polarity item and the NPI is still licensed, and NPIs may also be licensed through pragmatic factors (Xiang et al., 2009).



initiates a search by the parser for a negative polarity licenser that also c-commands the NPI. Consider the following examples from Vasishth et al. (2008):

(38) *No man* who had a beard was *ever* thrifty.

(39) \**A man* who had *no* beard was *ever* thrifty.

(40) \**A man* who had a beard was *ever* thrifty.

In (38) the negative DP *no man* c-commands the NPI *ever*. In (39) the negated DP *no beard* does not c-command the NPI because it is embedded in an RC and is therefore not syntactically accessible. In sentence (40) there is no negative context at all. Hence, in (39) and (40) the NPI *ever* does not occur in a negative context and is not licensed. Both sentences should be judged unacceptable because they violate the c-command constraint on NPIs.

Empirical studies in English have shown that sentences like (39) and (40) are indeed judged unacceptable in offline tasks. However, in online tasks, (39) is processed as if it was acceptable, exhibiting an *intrusion effect* (Drenhaus et al., 2005). This effect has been argued to reflect a memory access error which leads to the illusion during online processing that the sentence is temporarily acceptable even though it is ungrammatical (Vasishth et al., 2008; Wagers et al., 2009).

Drenhaus et al. (2005) investigated this intrusion effect in a speeded acceptability judgment task and an ERP experiment in German with the same three conditions as shown in the examples (38)–(40). The findings of the speeded acceptability task show that (39) was judged as being acceptable more often than (40) because of the intrusive licenser in the embedded clause (*no beard*). The results of the ERP experiment show an N400 effect in response to the illicit NPI for the ungrammatical conditions (39) and (40) with a reduced amplitude of the N400 for (39) compared to (40). The intrusion effect has been replicated in German using eye-tracking (Vasishth et al., 2008) and in English using ERPs (Xiang et al., 2009).

Vasishth et al. (2008) simulated the intrusion effect in the cognitive architecture ACT-R and argue for a partial matching of retrieval cues and memory encodings. Their account assumes that reading the NPI triggers a search for an item in memory with two properties: [+negative] and [+c-command]. They explain the intrusion effect by a partial matching which erroneously retrieves an item with the feature [+negative] leading to a temporary illusion of a licensed NPI.

All studies have agreed on the sensitivity of a retrieval mechanism to structural properties of a potential licenser. This sensitivity might vary depending on the time a comprehender has to process the NPI. More time as in offline tasks yields a higher possibility of a correct judgment while less time as in online experiments leads to an illusion of grammaticality.

### **3.3 Structurally-mediated Reactivation**

This section outlines studies on the resolution of pronouns and reflexives. Previous work suggests that the reactivation of the potential antecedent during online processing of pronouns and reflexives is structurally-mediated by binding constraints. This empirical evidence supports our argument that the human parser is influenced by structure rather than the mere amount of lexical material in a sentence.

#### **3.3.1 Pronoun and Reflexive Resolution**

Reactivation of potential antecedents to resolve a dependency relation has been found for sentences containing *pronouns* and *reflexives*. Binding theory suggests that the reactivation of the antecedent is guided by syntactic binding constraints and the detailed hierarchical structure of the sentence during online parsing (Dillon et al., 2014; Nicol and Swinney, 1989; Slattery et al., 2013).

Pronouns and reflexives are governed by complementary principles of binding theory (Chomsky, 1981). For reflexives the an-

tedecedent has to be the subject of the same clause or local domain as in (41) which is called *Principle A* in the classic binding theory (Chomsky, 1981). In terms of the Government and Binding (Chomsky, 1981) *Peter* c-commands *himself* and is therefore a grammatically appropriate antecedent. *Principle B* defines constraints on pronouns by ruling out a local binding as in (42) where *John* but not *Peter* can be the antecedent of the pronominal *him*. Again, *Peter* c-commands *him* and is therefore prevented from being the grammatically appropriate antecedent.

(41) John<sub>i</sub> thinks that Peter<sub>j</sub> hates himself<sub>\*i/j</sub>.

(42) John<sub>i</sub> thinks that Peter<sub>j</sub> hates him<sub>i/\*j</sub>.

Most studies have shown that the parser obeys the syntactic constraints and is sensitive to the completion of structural dependencies (Nicol and Swinney, 1989; Sturt, 2003; Xiang et al., 2009). The key difference between Principle A and B is that Principle A does determine the unique antecedent for a reflexive while Principle B merely rules out local binding and requires additional information sources (e.g., morphological, semantic and discourse-level properties) to determine the correct referent for a pronoun.

Although the studies agree on the syntactic constraints obeyed by the parser, there is a debate over the exact time-course of the processing of reflexives and pronouns. The question is whether processing is conditioned by structural constraints quite early (Nicol and Swinney, 1989) or whether features such as gender and number of the grammatically inappropriate antecedent still affect processing (Badecker and Straub, 2002).

To investigate this question Sturt (2003) conducted two eye-tracking experiments on the time-course of processing reflexives and focused on testing the online processing of the binding constraint Principle A. Sturt investigated to what extent ungrammatical antecedents influence the processing of binding constraints and to what extent the binding principles affect the final interpretation of a sentence, i.e., how often do people generate a final

interpretation that violates binding constraints. Sturt (2003) used the following four experimental conditions (43)–(46):<sup>4</sup>

(43) *Accessible-match/inaccessible-match*

*Jonathan* was pretty worried at the City Hospital. *He* remembered that the surgeon had pricked **himself** with a used syringe needle. There should be an investigation soon.

(44) *Accessible-match/inaccessible-mismatch*

*Jennifer* was pretty worried at the City Hospital. *She* remembered that the surgeon had pricked **himself** with a used syringe needle. There should be an investigation soon.

(45) *Accessible-mismatch/inaccessible-mismatch*

*Jonathan* was pretty worried at the City Hospital. *He* remembered that the surgeon had pricked **herself** with a used syringe needle. There should be an investigation soon.

(46) *Accessible-mismatch/inaccessible-match*

*Jennifer* was pretty worried at the City Hospital. *She* remembered that the surgeon had pricked **herself** with a used syringe needle. There should be an investigation soon.

In the first sentence of all conditions the first character is introduced with a proper name (*Jonathan* or *Jennifer*) and this character is referred to in the second sentence with a pronoun (either *he* or *she*). The subsequent sentence introduces a second character to the discourse *the surgeon* followed by a reflexive (*himself* or *herself*). Even though the first character is in the discourse focus here, it is not a grammatically appropriate antecedent for the reflexive (*inaccessible antecedent*) according to binding theory whereas the second character *the surgeon* is a possible antecedent (*accessible antecedent*). Sturt (2003) manipulated the gender agreement between the reflexive and the accessible and inaccessible antecedents. The

---

<sup>4</sup>Note: The labels for Sturt's conditions were corrected here for (45) and (46).

second character always had a stereotypical gender (*the surgeon* being stereotypically male).

Since Sturt (2003) was particularly interested in the timing of the effect due to the false binding of the inaccessible antecedent he put this antecedent into focus by introducing it with a proper name and referring to it with a pronoun in the next sentence. Additionally, the inaccessible antecedent (*he* or *she*) c-commanded the reflexive anaphor (*himself* or *herself*). The processing difficulty is predicted when the stereotypical gender of the accessible antecedent does not match the gender of the reflexive (as in (45) and (46)) compared to the conditions in which the gender does match ((43) and (44)).

Sturt (2003) showed that the retrieval mechanism of the parser is immediately influenced by the binding constraints on the antecedents for the reflexive. Early eye-tracking measures (e.g., first-fixation and first-pass reading times) that are usually associated with lexical and possibly structural processing showed a slowdown when the gender of the anaphor did not match the stereotype of the accessible antecedent (e.g., *surgeon ... herself* as in (45) and (46)) compared to when the gender matched (e.g., *surgeon ... himself* as in (43) and (44)). The gender of the inaccessible antecedent did not influence the early eye-tracking measures. But, in later eye-tracking measures (e.g., second-pass reading times) that are assumed to reflect integration of discourse information Sturt (2003) found an intrusion effect induced by the gender of the inaccessible antecedent. When the stereotypical gender of the inaccessible antecedent mismatched the gender of the reflexive (e.g., *Jennifer ... surgeon ... himself* as in (44)) the re-reading times were longer compared to the condition in which the stereotypical gender of the inaccessible antecedent matched the reflexive's gender (e.g., *Jonathan ... surgeon ... himself* as in (43)). To summarize, binding constraints immediately affect lexical access, but the inaccessible antecedent also has a delayed (discourse-based) effect during processing.

In a second experiment Sturt (2003) ruled out the possibility that the observed effects were driven by the linear position of the antecedents in the sentence. Cunnings and Felser (2013) also dissociated the effects of Principle A from effects of working memory limitations, i.e., linear distance, for reflexives. They manipulated the distance of the inaccessible and accessible antecedent in two eye-tracking experiments. In the first experiment Cunnings and Felser used similar materials to Sturt (2003) with three sentences, two characters in the discourse and a reflexive in the four experimental conditions (43–46) and gender congruence between the reflexive and the two characters. Cunnings and Felser (2013) were further interested in how readers' working memory (WM) capacity interacts with antecedent effects. Since the representation of an antecedent decays in memory while reading the sentence, Cunnings and Felser predicted that participants with low WM span may be more affected by decay than high WM span readers. Hence low WM participants are more insensitive to the gender manipulation of the nonlocal inaccessible antecedent and might try to keep the dependency between the reflexive and the antecedent short. High WM span readers might be less affected by decay and show effects of the more distant inaccessible antecedent. In sum, according to Cunnings and Felser (2013) low WM span readers are expected to show more pronounced and earlier effects of the stereotypical gender manipulation of the linearly closest accessible antecedent compared to high WM span readers.

The results of their first experiment showed an effect of WM capacity. The low WM span group showed an earlier effect of the accessible antecedent mismatch, i.e., longer first fixation and regression path durations in the reflexive region, than the high WM span group who showed longer rereading times in the reflexive region. Effects of the inaccessible antecedent during were found in later stages of processing. But, Cunnings and Felser (2013) point out, the early effect of the accessible antecedent mismatch might reflect that applying Principle A is simply a "last resort" constraint instead of being syntactically constrained independent of linear

distance between the antecedent and the reflexive. Therefore, Cunnings and Felser (2013) conducted a second eye-tracking experiment in which the order of the two antecedents was reversed compared to the ordering in experiment 1 (see also 43–46). Again, both groups showed an intrusion effect of the structurally accessible antecedent, with longer rereading times at the reflexive and spillover regions and longer regression path times in the spillover region when the reflexive mismatched the stereotypical gender of the antecedent. Surprisingly, only the low WM span group showed early effects of the linearly closer inaccessible antecedent in longer first fixation durations and longer first-pass times when the gender of the inaccessible antecedent mismatched the gender of the reflexive. From the fact that only low WM span readers were affected early by both antecedents, Cunnings and Felser (2013) conclude that Principle A competes with recency or discourse prominence of the structurally inaccessible antecedent in WM. Readers need to inhibit the inaccessible antecedent or retrieve the less activated more distant accessible antecedent from WM. Low WM readers may have had more difficulty with the inhibition or retrieval process than high WM readers and consider both antecedents as a potential candidate while linking to the reflexive early during processing.

Similarly, Clackson et al. (2011) tested if adults and children in a visual world listening experiment simply favored the linearly closest antecedent or the most prominent one in the discourse. Their results corroborate the findings of Cunnings and Felser (2013), the eye gaze patterns show clear preferences for the structurally accessible antecedent according to binding principles for both pronouns and reflexives. They also found interference effects from the inaccessible antecedent in both groups suggesting that the gender-matched inaccessible competitor still caused distraction. To summarize, all these studies suggest a structurally-mediated reactivation of potential antecedents to resolve the dependency for pronouns and reflexives in online sentence comprehension.

## 3.4 Computational Models of Structural-based Approaches

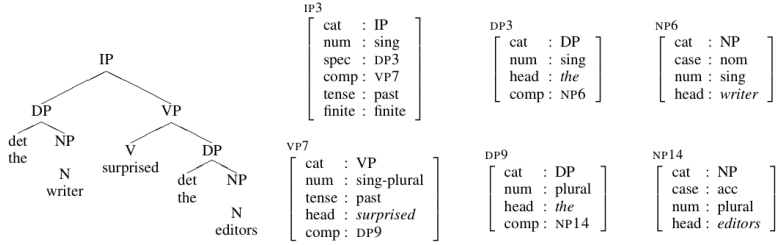
The following section will review a few computational models implementing structural-based approaches. These models link syntactic analysis provided by a grammar theory to human behavior. They work incrementally on the input, use grammatical knowledge to guide the parsing process, and implement some notion of working memory, e.g. activation decay. The computational models can account for various empirical data of syntactic complexity.

### 3.4.1 ACT-R

ACT-R is a computational model that has been used for syntactic parsing to model empirical data in sentence comprehension (Lewis and Vasishth, 2005; Vasishth et al., 2008). The model combines memory representations and grammatical knowledge in the form of production rules (Anderson et al., 2004). The working memory component of ACT-R is composed of a *limited focus of attention* and *declarative memory elements* with a certain level of activation depending on the time since these elements are created or processed. The activation of a memory element starts to decay immediately after it is processed. The memory elements in the focus of attention determine processing since the production rules are only matched against *chunks*, i.e., linguistic constituents, in a limited set of buffers (Vasishth et al., 2008). The set of chunks contain lexical elements that are represented as feature-value specifications as illustrated in Figure 3.2. These lexical elements are either stored in memory or held in an active state in order to build syntactic structure. A sentence like *The writer surprised the editors.* is a set of chunks representing either terminals or non-terminals and production rules build a parse tree as in Figure 3.2 (Vasishth et al., 2008).

The parser constitutes a left-corner parsing strategy (Lewis and Vasishth, 2005; Vasishth et al., 2008). Memory access or retrieval





**Figure 3.2:** Lexical elements<sup>5</sup> in ACT-R's memory corresponding to the sentence *The writer surprised the editors.* Lewis and Vasishth (2005, p. 6)

works by using certain retrieval cues to access chunks in a content-addressable fashion (cue-based retrieval). ACT-R is a model of working memory that focuses “on retrieval interference and activation decay as the factors limiting performance, rather than storage capacity” (Van Dyke and Lewis, 2003, p. 286).

A number of studies used ACT-R to model cue-based retrieval in sentence comprehension (Lewis and Vasishth, 2005; Lewis et al., 2006; Vasishth et al., 2008) and the performance of ACT-R can explain various empirical findings in English (Lewis and Vasishth, 2005), German (Vasishth et al., 2008) and Hindi (Vasishth and Lewis, 2006). The assumption underlying these simulations of data is that there is a monotonic relation between reading times, which partly reflect the timing of retrieval operations in online comprehension, and the retrieval latencies predicted by the model that are affected by the activation level of the retrieved chunk. Additionally, Vasishth et al. (2008) used the *partial matching* functionality of ACT-R to simulate the intrusion effect of a structurally inaccessible licenser of polarity items. Partial matching is possible if the retrieval specification matches only a part of the chunk's feature values and its activation remains above a certain threshold; then the chunk will be retrieved successfully.

<sup>5</sup>IP = inflectional phrase; DP = determiner phrase; VP = verb phrase; NP = noun phrase; V = verb; N = noun; det = determiner.

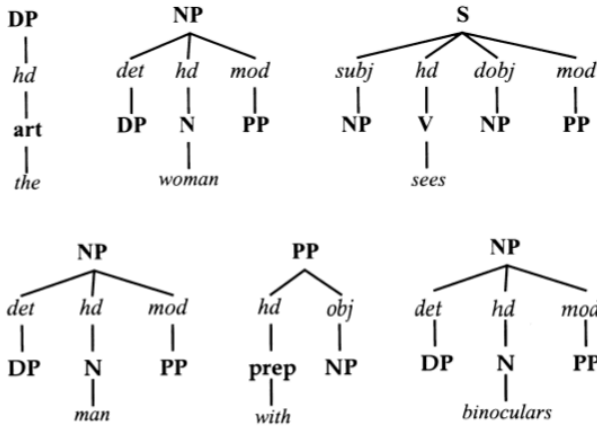


Figure 3.3: Lexical frames<sup>6</sup> in the Unification Space model corresponding to the sentence *The woman sees the man with the binoculars*. Vosse and Kempen (2000, p. 110)

### 3.4.2 The Unification Space Model

Another psycholinguistically motivated model of human sentence comprehension is the *Unification Space model* proposed by Vosse and Kempen (2000, 2009). The parser of this model is based on a lexicalized grammar and the algorithm is driven by syntactic information associated with the head of the phrase. The parser creates and operates on *lexical frames* that are retrieved as *chunks* from the mental lexicon. Each connection between chunks represents a possible attachment alternative. The strength of these connections varies depending on local competition of potential alternatives. The more alternative<sup>6</sup> analyses exist the harder the parse of the sentence becomes. If the syntactic analysis of the input sentence is successful, then the parser will output a single grammatically correct syntactic tree. Figure 3.3 shows the lexical frames in the lexicalized grammar. There are no syntactic rules. The top layer of

<sup>6</sup>DP = determiner phrase; NP = noun phrase; PP = prepositional phrase; S = sentence; N = noun; V = verb; art = article; prep = preposition; hd = head; det = determiner; mod = modifier; subj = subject; dobj = direct object; obj = object.

a lexical frame is a single *phrasal* node which is connected to one or more *functional* nodes (Vosse and Kempen, 2000). Each functional node dominates a third layer; here a head (hd) immediately dominates a lexical node. Lexical nodes dominate words. Further, each lexical node is associated with a feature matrix specifying grammatical gender, person, number and case.

The function that the model uses to combine lexical frames is called *unification*. Two conditions need to be fulfilled before two nodes can be unified. First, the grammatical features of both nodes must be compatible, i.e., nodes must agree in number, person and case. Second, word order must be retained, i.e., “a root node of a lexical frame is allowed to unify with a foot node of another frame only if this does not violate a precedence rule for the branch dominating the foot node.” (Vosse and Kempen, 2000, p. 111).

As soon as a node of a lexical frame enters the Unification Space, i.e., it is a possible candidate for the syntactic structure, it receives a certain *activation* value which initially comes from the mental lexicon and depends on the frequency of usage of the lexical frame. During the parsing process the activation gradually decays towards 0 if the node is not reactivated.

The Unification Space model can explain a number of phenomena in English such as complexity (e.g., center-embedded versus left-branching versus right-branching structures; SRCs versus ORCs; contrast between a complement clause embedded within an RC versus an RC embedded within a complement clause), syntactic ambiguity (e.g., global and local attachment ambiguities; frequency differences between attachment ambiguities), lexical ambiguity (including recency and length effects), certain difficulty-of-reanalysis effects (local ambiguities that induce weak or severe garden-path effects) and even effects of agrammatism on parsing performance such as data from aphasic patients (Vosse and Kempen, 2000).

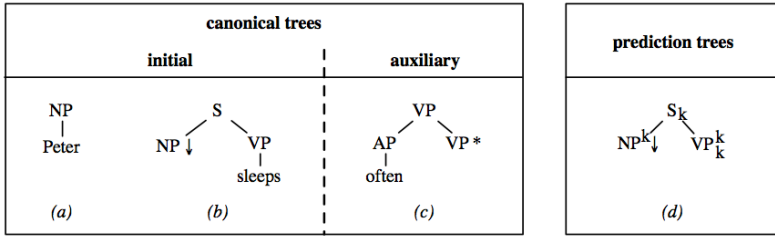


Figure 3.4: Elementary trees<sup>7</sup> in the lexicon of PLTAG corresponding to the sentence *Peter often sleeps*. Demberg et al. (2013, p. 1030)

### 3.4.3 The Prediction Theory

Quite recently, Demberg et al. (2013) proposed the Prediction Theory based on a Psycholinguistically Motivated Tree Adjoining Grammar (PLTAG (Demberg and Keller, 2009)). PLTAG works incrementally on fully connected structures and makes predictions about upcoming words. The formalism extends a standard Tree Adjoining Grammar (TAG) developed by Joshi et al. (1975) with a predictive lexicon and a verification operation (Demberg et al., 2013). TAG operates on *initial* and *auxiliary* trees as illustrated in Figure 3.4. Initial trees contain a *substitution node* labeled with a symbol  $X\downarrow$  and auxiliary trees contain a *foot node* labeled with the symbol  $X^*$ . The two tree-building operations of TAG are *substitution* and *adjunction*. Substitution combines the two trees for *Peter* and *sleeps* while adjunction integrates the auxiliary tree of *often*, deriving the syntactic tree for *Peter often sleeps*.

Additionally, PLTAG defines *prediction trees* (see right part of Figure 3.4). These trees are used to predict syntactic structure for later words in the sentence. The markers  $k$  and  $k$  on the nodes of the prediction tree indicate that they are not actually part of the syntactic structure but only predicted nodes. Prediction trees serve as temporary trees to enable an incremental parsing routine, and they will be substituted later by an elementary tree. For instance, to incrementally parse *Peter often sleeps*, the tree *Peter* is

<sup>7</sup>NP = noun phrase; S = sentence; VP = verb phrase; AP = adverbial phrase.

combined with the prediction tree (d) and will adjoin to the tree *often*. Now, the prediction tree will be substituted by the initial tree for *sleeps*. Next, the *verification* operation removes the markers of the prediction tree.

The PLTAG lexicon is derived from the Wall Street Journal section of the Penn Treebank which was converted to a PLTAG format (Demberg et al., 2013). Prediction Theory, proposed by Demberg and Keller (2008a, 2009), calculates word-by-word difficulty scores from the PLTAG parser. Difficulty scores are computed as the sum of two components: surprisal (Hale, 2001) and verification cost. Surprisal is calculated from the probability distribution over prefix trees spanning the input. Verification cost depends on the probability of a prediction tree and on memory decay, i.e., distance between prediction and verification (Demberg et al., 2013). Hence, Prediction Theory combines two types of complexity metrics: “The cost of updating syntactic representations (surprisal) and the cost of integrating predicted structure (verification cost)” (Demberg et al., 2013, p. 1053).

PLTAG correctly predicts reading time data which capture SRC versus ORC asymmetries and *either ... or* structures from Staub and Clifton (2006) (Demberg and Keller, 2009). For verification cost alone no significant effect overall is observed, which means that verification cost does not make predictions on the majority of words in the corpus. Demberg et al. (2013) suggest that most verification cost values are zero as no verification is necessary at most words. However, in combination with surprisal, the effect for verification cost is improved.



## 4 The Parsing Algorithm

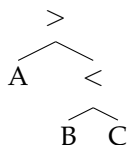
This chapter will outline the architecture of the implemented parsing algorithm. The parser is based on the grammar formalism minimalist grammars (Stabler, 1997) which will be introduced in Section 4.1. Section 4.2 will discuss different parsing strategies, namely bottom-up, top-down and left-corner parsing. Finally, in Section 4.3 the components of the implemented minimalist top-down parser will be outlined: (1) the grammar fragment based on a minimalist grammar proposed by Hale (2003b); (2) the implemented top-down parsing algorithm working on the grammar fragment following theoretical work by Mainguy (2010); and (3) the complexity metric (related to work by Rambow and Joshi (1994) and Kobele et al. (2013)) which derives online and offline complexity profiles for sentence structures.

### 4.1 The Minimalist Grammar

The implemented algorithm presented in this chapter parses languages generated by *minimalist grammars*, as formalized and defined by Stabler (1997) (see also Stabler, 2011a). Stabler formalized the kind of grammars proposed by Chomsky's *Minimalist Program* (Chomsky, 1995).

Minimalist grammars are derivational and feature-driven: words are combined into phrases through transformational operations applied to words and intermediate structures. The relevant operations are triggered by certain syntactic features. Words or lexical items are combined into trees. Unlike in the traditional Government and Binding theory where the root of a tree is labeled with the label of a head or a phrase, e.g. V, VP, NP, PP, trees built

by minimalist grammars have a projection indicator as the root of the tree pointing to the head of the projection, as in Figure 4.1:



**Figure 4.1:** A tree built by a minimalist grammar

The pointers in Figure 4.1 lead to a head labeled “B” thus the structure is a projection of B.

A minimalist grammar uses words or lexical items to build trees. Lexical items contain different types of features: *syntactic features* and *non-syntactic features* (e.g., phonetic features and semantic features). Since this thesis is about syntactic parsing, only syntactic features will be discussed.<sup>1</sup> Syntactic features can be of four types: (1) *categories* representing the syntactic category of the lexical item, e.g. n for noun, v for verb; (2) *selectors* representing a syntactic category preceded by a “=”, e.g. =n for “selects noun”, =v for “selects verb”; (3) *licensees* represented by a letter preceded by “-”, e.g. -wh, -f; (4) *licensors* represented by a letter preceded by “+”, e.g. +wh, +f.

The following toy grammar will be used to build the syntactic structure of the sentence *The manager hired the employee*. An empty lexical item is written as  $\epsilon$ .

- *the* :: =n d -case
- *manager* :: n
- *hired* :: =d +case =d +case t
- *employee* :: n
- $\epsilon$  :: =t c

---

<sup>1</sup>See Stabler (1997) for a discussion of syntactic and non-syntactic minimalist features.



### 4.1.1 Structure Building Operations

The two structure building operations of minimalist grammars are *merge* and *move* (Stabler, 1997). Merge combines lexical items or intermediate structures using categories (e.g., *n*, *v*) with selectors (e.g., *=n*, *=v*). Merge can be applied to two different subtrees: *simple* and *complex* trees. A tree with more than one node as shown in Figure 4.3 is called a *complex tree*. Lexical items are *simple trees*. A simple tree will always take a new sister to its right, while a complex tree will take a new sister to its left (Harkema, 2001).

Move applies to trees whose left-most feature of the head is a licenser (e.g., *+wh*, *+case*) and which contain another head with a matching licensee feature (e.g., *-wh*, *-case*). The constituent with the licenser will be moved upwards in the tree to a specifier position of the head with the licensee feature, leaving a co-indexed trace behind to mark the original position. Note that the move operation is restricted by the Shortest Movement Constraint (SMC) (Stabler, 1997): Move can only apply to a tree which has exactly one head with a licensee feature matching a corresponding licenser in the same tree. For instance a tree like the one in Figure 4.2 violates the SMC, because it contains two subtrees with the licensee feature “*-f*” that compete for the same specifier position of “*+f*”. This tree is not in the domain of the move operation.

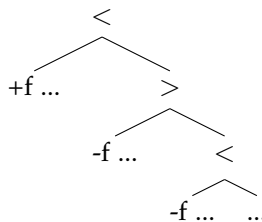


Figure 4.2: A tree violating the Shortest Movement Constraint

An example of the merge operation is shown in Figure 4.3. The determiner “*the* :: · =*n* d -*case*” and the noun “*employee* :: · *n*” are merged. The dot · is used to keep track of derivations, and it is

placed in front of the *current feature* (Mainguy, 2010). Here, the category “n” and the selector “=n” trigger the merge operation. The syntactic features of both lexical items are copied to the root of the new tree and the dot moves one position to the right for each lexical item. Since there are no more syntactic features left for *employee* the syntactic features for this word are omitted in the next derivations. Hence, the resulting rule is “=n · d -case” and the projection indicator “<” in root position of the tree points to the head of the phrase, the determiner *the*. The lexical item that contained the selector will always become the head of the phrase and is written as the first rule in root position.

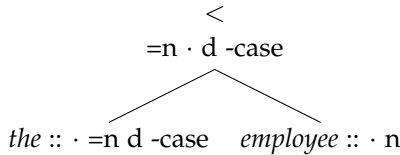


Figure 4.3: Merge *the* and *employee*

Next, “*hired* :: · =d +case =d +case t” is merged to the subtree of *the employee* as shown in Figure 4.4.

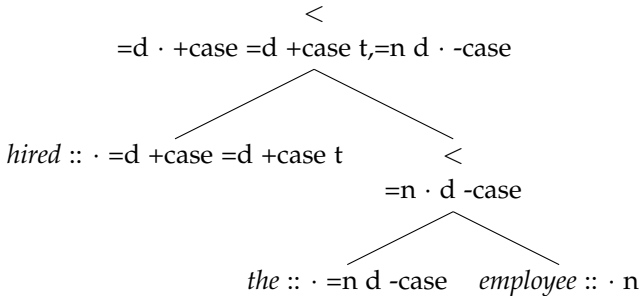


Figure 4.4: Merge *hired* and *the employee*

This time the syntactic features of *hired* are not yet processed, so the dot moves one position to the right and the features of *hired* are written before the features of the determiner phrase *the employee* in root position of the resulting tree.

Now, the move operation is triggered by the features “+case” and “-case” in the tree. The constituent *the employee* is moved upwards in the tree because it is the constituent with the licensor “+case” (as in Figure 4.5).

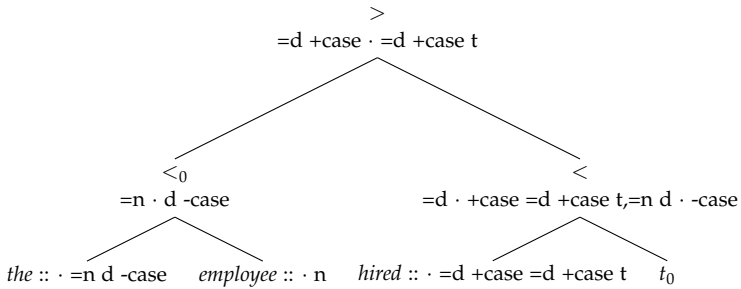


Figure 4.5: Move *the employee*

The dot moves to the right of “+case” in the root node and the second rule is omitted because there are no syntactic features left for it. The projection indicator “>” in root position points to the branch of the tree where the licensor used to be. As shown in Figure 4.5 the constituent *the employee* leaves a trace “ $t_0$ ” to mark its original position.

Next, *the manager* is merged to the subtree of *the employee hired* as shown in Figure 4.6.

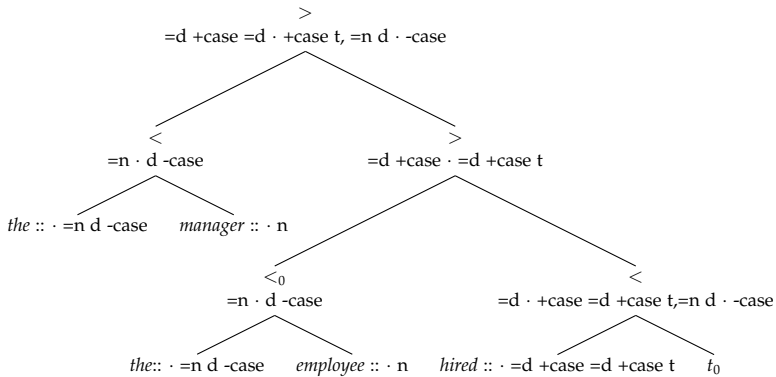


Figure 4.6: Merge *the manager*

The dot moves to the right in both rules and the syntactic features of the head *hired* are written first.

In Figure 4.7 move is applied to the tree, this time moving *the manager* to the top of the tree:

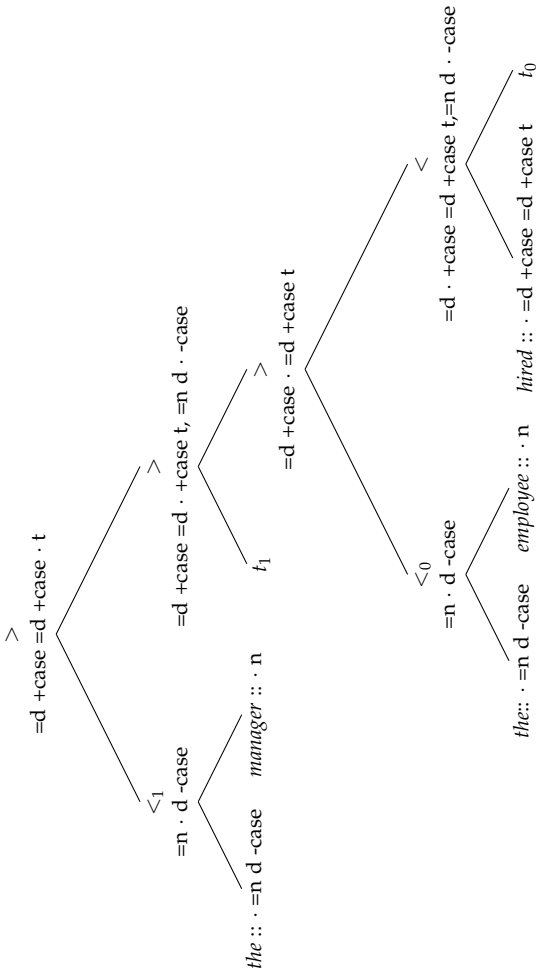


Figure 4.7: Move the manager

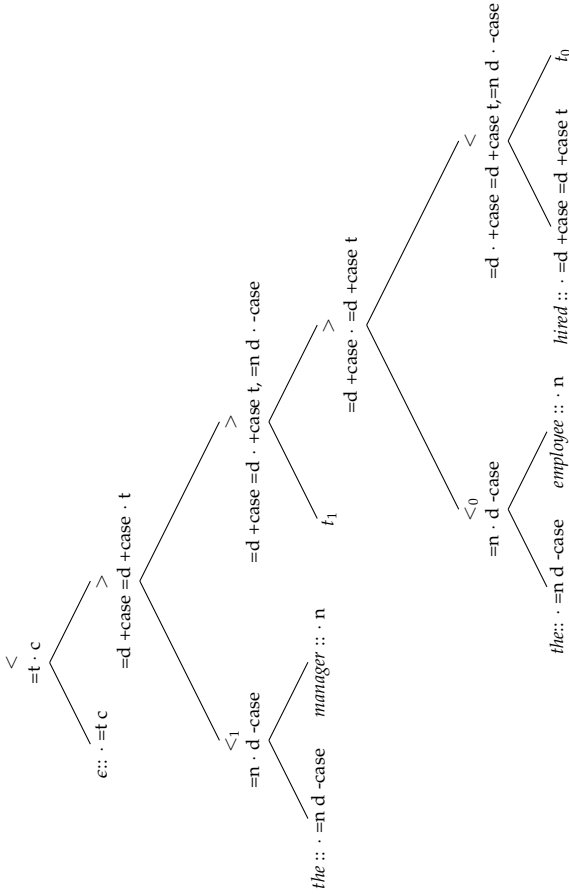


Figure 4.8: Merge  $\epsilon$  to the manager the employee hired

Finally, “ $\epsilon :: \cdot =t c$ ”, an empty lexical item, is merged to the tree as shown in Figure 4.8.

The only syntactic feature to the right of the dot is the distinguished feature “ $c$ ”. This feature represents a complementizer that finalizes the derivation of the sentence and yields a grammatical tree.

### 4.1.2 Derivation Trees

A different way of representing the structure of a sentence built by a minimalist grammar is a *derivation tree* (Kobele, 2006; Mainguy, 2010). A derivation tree shows the history of the applied structure building operations, merge and move. Its leaves are lexical items. Following the notation of Mainguy (2010) binary nodes represent a merge operation and are labelled  $\bullet$ , unary nodes represent move and are labelled  $\circ$ . The derivation tree for the example *The manager hired the employee* in Figure 4.8 can be rewritten as:

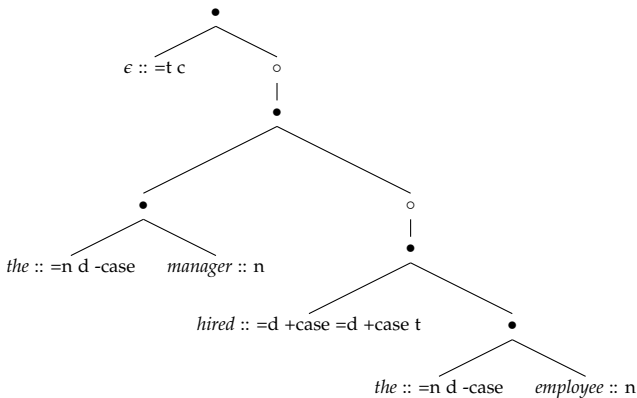


Figure 4.9: Derivation tree of *The manager hired the employee*

A derived tree (as in Figure 4.8) can be easily computed from such a derivation tree (Kobele, 2006; Kobele et al., 2007; Stabler, 2011b). Derived trees reflect the surface structure of the sentence, i.e., the order of the words, while derivation trees give the history

of the structure building operations. Therefore the geometry of a derivation tree does not always coincide with the order of words in the derived tree (Harkema, 2001). The following section will discuss parsing strategies which can parse derivation trees in two different ways, from bottom to top and from top to bottom, to construct derived trees.

## 4.2 Parsing Strategies

### 4.2.1 Bottom-up and Top-down Parsing

Parsing can be considered as the syntactic analysis of words in a sentence and their combination into a syntactic structure following the rules of a grammar. The human parser constructs a parse tree of an input incrementally, word-by-word (Marslen-Wilson, 1973; Phillips, 2003; Tanenhaus et al., 1995). A parse tree is built *online* while processing each word (Frazier and Fodor, 1978; Just and Carpenter, 1980; Pickering et al., 1999). Hence, it is interesting to study the computational resources of parsing algorithms (Abney and Johnson, 1991).

Kimball (1973) is one of the first who reviews *parsing strategies* from a computational perspective. He describes *bottom-up* parsing, in which the structure for the sentence is built starting with the categories of the words up to the top of the syntactic tree, and *top-down* parsing, in which the parser builds the structure starting with the top-most symbol going down to the words.

As said before, a derivation tree derived by a minimalist grammar, such as the one in Figure 4.9, can be parsed in these two ways: bottom-up and top-down (Harkema, 2001).<sup>2</sup> A *bottom-up parser* starts at the bottom right of the tree by merging the non-terminals *the* and *employee*. Then, *hired* is merged to the constituent *the employee* and so forth. In this way the parser works its way up the derivation tree by repeatedly applying merge and

---

<sup>2</sup>Moreover, if weights are appointed to the branches, derivation trees can be viewed as probabilistic context-free grammars (Hale, 2003b).



move until the distinguished category  $c$  is merged to the syntactic structure finalizing the derivation. Harkema (2001) describes such a bottom-up recognizer for minimalist grammars similar to the Cocke-Younger-Kasami (CYK) algorithm for context-free grammars (Younger, 1967).

Syntactic trees for minimalist grammars are usually generated bottom-up by merging and moving constituents (Stabler, 1997). To complement this, Harkema (2001) and Mainguy (2010) developed a rewriting system for minimalist grammar rules working top-down by using *inference rules*. For each merge and move operation of the minimalist grammar there is a corresponding inference rule *un-merge* and *un-move* which applies the original operation in reverse. A *top-down parser* for the derivation tree in Figure 4.9 would start at the top of the tree and work its way down to the leaves. At first, the parser applies un-merge to the rule " $\epsilon :: =t c$ " and adds another rule to the *relevant categories* of the parser to apply un-move in the next step. The relevant categories of the parser determine the set of rules that the parser has to process. When the parser reaches a leaf, the lexical material will be scanned, i.e., the parser will match the lexical item at the leaf with the words in the input sentence. The parse is finished as soon as the algorithm reaches the last leaf at the bottom right of the derivation tree.

### 4.2.2 Left-corner Parsing

The parsing strategies, bottom-up and top-down parsing, yield problems with certain syntactic structures. While a bottom-up parser is able to process left-branching structures, it has problems with right-branching structures (Miller and Chomsky, 1963). Even though a top-down parser can process right-branching structures, it has problems with left-branching structures (Miller and Chomsky, 1963). According to Abney and Johnson (1991) neither of the two parsing strategies has the desired property to predict memory requirements correctly for center-embedded structures. Because of these problems with the two parsing strategies, the human processor is assumed to use a mixed strategy of top-down predictions

and bottom-up confirmations, such as a *left-corner* algorithm (Aho and Ullman, 1972; Johnson-Laird, 1983; Resnik, 1993).

A left-corner parser processes the left-most category of the right hand side of a grammar rule bottom-up and the rest of the grammar rule top-down. The following simplified example illustrates left-corner parsing. Consider the following grammar rules:

1.  $S \rightarrow NP VP$
2.  $NP \rightarrow Det N$
3.  $VP \rightarrow V$
4.  $Det \rightarrow the$
5.  $N \rightarrow boy$
6.  $V \rightarrow sleeps$

The input sentence is *The boy sleeps.* and the left-corner parser starts with rule 1 by rewriting the symbol “S” into “NP VP”. The parser proceeds with the left-hand side of the rule until no further replacements are possible, i.e., the parser expands “NP” into “Det N” (rule 2) and “Det” is rewritten as *the* (rule 4). Next, the parser expands “N” to *boy* (rule 5), “VP” is expanded to “V NP” (rule 3) and finally “V” is rewritten as *sleeps* (rule 6).

According to Resnik (1993) a left-corner parser can explain center-embedded structures using a context-free grammar. But, a left-corner parser, which uses a mildly context-sensitive grammar formalism such as minimalist grammars, is either ill-defined or not available yet (Kobele et al., 2013). Therefore, the implemented parser in this thesis uses the top-down strategy as an alternative parsing strategy to simulate human parsing behavior and incremental, word-by-word sentence processing on connected structures. The parser is an implemented version of Mainguy’s 2010 parser. Mainguy (2010) not only describes a (probabilistic) top-down parser for minimalist grammars but he also conducted a proof of its soundness and completeness.

### **4.3 The Components of the Implemented Minimalist Top-down Parser**

Several incremental parsers based on grammar formalisms have been suggested in the previous literature. For instance, Roark (2001) proposed an incremental parser based on context-free grammars. The parser implements a top-down algorithm with fully connected syntactic structures and computes probabilities for sentence prefixes. The parser successfully predicts reading time data (Demberg and Keller, 2008b; Roark et al., 2009). Nivre (2004) developed an incremental parser based on a dependency grammar and Boston et al. (2008) show that the parser successfully models data from an eye-tracking corpus in German. Demberg et al. (2013) developed an incremental, predictive parser based on a Lexicalized Tree Adjoining Grammar called PLTAG which predicts reading times from the Dundee eye-tracking corpus in English.

Minimalist grammars can be parsed bottom-up, building a syntactic structure starting from the lexical entries (Stabler, 1997; Hale, 2003b; Harkema, 2001). However, a bottom-up parser does not have the predictive power of a top-down parser, hence several top-down parsing algorithms for minimalist grammars have been proposed (Harkema, 2001; Mainguy, 2010; Stabler, 2011b, 2013). For instance, Harkema (2001) defined a top-down Early-style recognizer for minimalist grammars that uses a chart to keep track of all possible syntactic analyses. This thesis implements an incremental serial top-down parser for minimalist grammars described by Mainguy (2010) in combination with an intuitive complexity metric (Rambow and Joshi, 1994). The minimalist parser uses a head-driven strategy that introduces phrasal nodes by their heads (e.g., verb, noun, preposition), so that the attachment of pre-head constituents is postponed until the head arrives (Abney and Johnson, 1991; Vosse and Kempen, 2000).

The proposed top-down parser takes as input a sentence as a string  $\omega_0^n$  and a minimalist grammar fragment  $G$ . The parser works deterministically, meaning that it applies at most one

operation in one parse step (Marcus, 1980; Aho and Ullman, 1972). We use a so-called *oracle* because we are currently not concerned with factors related to non-determinism. While having a perfect oracle is not possible, we may think of it as an idealization of a good set of heuristics together with a narrow beam width. As output, the parser creates a complexity metric for the input sentence. The components of the parsing algorithm (the grammar fragment, the oracle of the parser, the top-down parsing algorithm and the calculation of the complexity metric) will be explained in more detail in the following sections.

### 4.3.1 The Grammar Fragment

The grammar fragment on which the incremental parser operates is a modified version of the “Larsonian” minimalist grammar (MG) developed by Hale (2003b). The Larsonian MG depicts the promotion analysis of relative clauses as defined by Kayne (1994). The grammar uses the traditional features of a minimalist grammar introduced in Section 4.1, e.g. categories, selectors, licensees and licensors. Additionally, Hale (2003b) extended the traditional features of merge and move to gain more flexibility in grammatical analyses.

In contrast to the standard adjunction analysis of relative clauses, the promotion analysis uses a non-concatenative move rule (Hale, 2003b). Consider the following lexical entries:

- (47) a. manager :: n
- (48) b. manager :: n -f
- (49) who :: =n +f d -case -wh

There are two entries for the common noun *manager*, one of them with the optional *promotion* feature -f (“focus”). If the input sentence depicts a relative clause with *who*, the movement of *manager* into the specifier position to the left of *who* can be triggered by the corresponding licensor feature “+f” of *who*. The moved constituent *manager* leaves a trace  $t_0$  behind. Movement reverses the

linear order of *manager* and *who* in the surface structure yielding *manager who*. If the input sentence is not a relative clause, then the alternative lexical entry for *manager* is used.

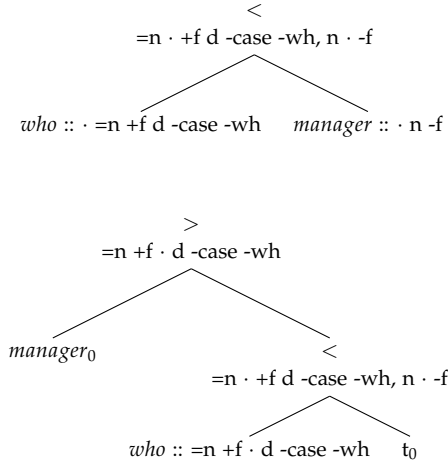


Figure 4.10: Move *manager* into specifier position to the left of *who*

Another arrangement of strings in the Larsonian MG which is implemented as a move operation is *case-checking*. Determiner phrases have a “-case” licensee feature while case-assigning categories such as verbs, prepositions and tense have a corresponding “+case” licenser feature (Haegemann, 1994; Hale, 2003b). For instance, prepositions can check case:

(50)  $\epsilon :: =>P_{\text{from}} p_{\text{from}}$

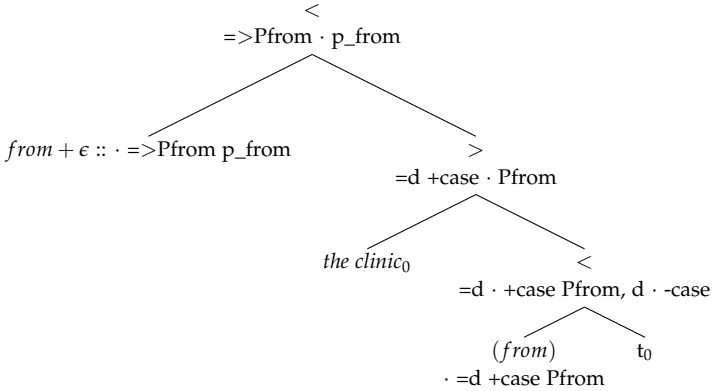
(51)  $be :: =p_{\text{from}} v$

(52)  $from :: =d +case P_{\text{from}}$

(53)  $the\ clinic :: d -case$

The preposition *from* selects a determiner phrase (=d) like *the clinic*. Then, the determiner phrase is moved upwards into the specifier position of *from* triggered by “-case” and “+case”. Then the rule “ $\epsilon :: =>P_{\text{from}} p_{\text{from}}$ ” is merged as a complement to the subtree.

The special symbol “=>” before “Pfrom” means *left incorporation* and incorporates the head of the prepositional phrase, *from*, to the left side of rule “ $\epsilon :: =>Pfrom p\_from$ ” (Hale, 2003b; Stabler, 2001). The resulting prepositional phrase can then be selected by a verb, e.g. *be*, to yield *be from the clinic*.



**Figure 4.11:** Left incorporation of *from* as head of prepositional phrase *from the clinic*

Following Hale (2003b), the Larsonian grammar implements *affix hopping* which applies head movement to combine tense affixes with verb stems, e.g. *stole* is segmented into *steal -ed*, *hired* into *hire -ed* etc. (see also Kobele, 2006; Stabler, 2001, 2011b).

- (54)  $hire :: =d +case v$
- (55)  $-ed :: =>v +case t$
- (56)  $\epsilon :: =>v =d v$
- (57)  $the\ nurse :: d -case$
- (58)  $the\ manager :: d -case$

To obtain the tree in Figure 4.12 below, at first, *hire* and the phrase *the nurse* are merged. Then the rule “ $\epsilon :: =>v =d v$ ” merges to *the nurse hire* and the verb *hire* hops before  $\epsilon$  to yield the phrase *hire the nurse*. After merging *the manager* to *hire the nurse* the affix

*-ed* is integrated into the structure and *hire* is incorporated as the head of the phrase *hired the manager the nurse*. The next step is to move *the manager* upwards in the tree to obtain the linear order for the clause *the manager hired the nurse*.

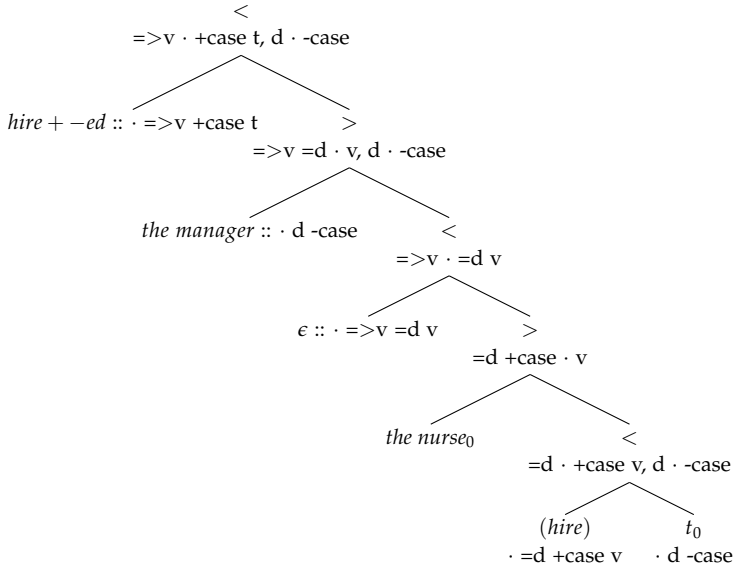


Figure 4.12: Affix hopping to combine tense affixes and verb stems

The complete grammar fragment used in the top-down parser to model the phenomena presented in this thesis is given in Appendix 7.

### 4.3.2 The Oracle of the Parser

The so-called oracle of the parser is a derivation tree constructed by a bottom-up parser called Cornell Conditional Probability Calculation (CCPC)<sup>3</sup> developed by John Hale, Tim Hunter, Zhong Chen and Jiwon Yun which uses the same grammar fragment as

<sup>3</sup>The parser is available at: <http://conf.ling.cornell.edu/compling/software.htm> (retrieved May 21, 2014).

the presented top-down algorithm. Additionally, the implemented top-down parsing algorithm and the oracle use the same sentence as an input string. The oracle computes a derivation tree in XML tree format which sometimes serves as a look-up for the top-down parser. Both parsers, the oracle and the top-down parser, generate an identical derivation tree only in two different ways (bottom-up vs. top-down). Each time a grammar rule can be expanded in two different ways, the top-down parser searches for the correct grammar rule for the input string in the derivation tree derived by the CCPC.

For instance, when the parser encounters the rule “=n · d -case”, it looks for a corresponding grammar rule with “n” as the right-most feature (for more details on how the algorithm works see Section 4.3.3). There are two possible grammar rules in the grammar fragment:

(59) n ·

(60) =ce n ·

Rule (59) predicts a common noun and rule (60) a noun that requires a complement, e.g. *fact* in *the fact that*. The parser determines the correct rule for the current input string by using the oracle as a look-up and uses this rule to continue processing the input string.

### 4.3.3 The Top-down Parsing Algorithm

The proposed parsing algorithm implements an incremental top-down parser, theoretically defined by Mainguy (2010) (see also Harkema, 2001, Stabler, 2011b, 2013). Mainguy (2010) proved the soundness and completeness of the parser. The following section will repeat the definitions by Mainguy (2010) and illustrate their functionality with an example.

The top-down parser uses a last-in first-out stack organized as a *priority heap* to keep track of the current derivations. Since move is possible, the parser has to keep track of the position of a constituent or word in the derived tree. The derived tree yields the



surface structure of the sentence. Hence, the order of rules is determined by *position indices*  $\alpha_0 \dots \alpha_k$  that denote positions of constituents in the derived tree through a series of digits. If  $\alpha_0 = 0$  the next subtree is obtained by going down to the left from its root, and  $\alpha_0 = 1$  means going down to the right from its root. At each parse step the priority heap is ordered left-to-right by ascending order of position indices. Always the left-most rule is expanded. In this way only the leaves of the currently parsed word are expanded. The assignment of position indices is determined by the *inference rules* which will be explained next.

#### 4.3.3.1 The Axiom

The axiom of the parser is the category *start* plus an empty position index  $\epsilon$  (Mainguy, 2010). The position index is written before “/” in the rule. For instance,  $\epsilon$  is written before *start* for the axiom ( $\epsilon/start$ ).

#### 4.3.3.2 The Inference Rules

Given a minimalist grammar  $G = (\sigma, \text{Cat}, \text{Lex}, F)$  where  $\sigma$  is a set of non-syntactic features,  $\text{Cat}$  is a set of syntactic features,  $\text{Lex}$  is a set of expressions built from  $\sigma$  and  $\text{Feat}$  (lexicon), and  $F$  is a set of generating functions (merge, move) (Stabler, 1997). The syntactic non-terminal symbols of grammar  $G$  are categories. A category is a sequence of the form  $[\gamma_0 \cdot \delta_0, \dots, \gamma_n \cdot \delta_n]$  with  $\gamma_j, \delta_j \in \text{Cat}^*$  ( $0 \leq j \leq n$ ) yielding a tree  $t \in \{c, s\}$ , where “ $c$ ” denotes a complex tree and “ $s$ ” indicates a simple tree (Harkema, 2001; Mainguy, 2010).

Following the definition of Mainguy (2010), there are two types of categories: a *simple* category is a category with a dot at the left-most place (and  $k=0$ ), otherwise a category is a *complex* category. The axiom *start* is neither simple nor complex.

According to Mainguy (2010) a *partial output* is a string  $\Delta_1 \dots \Delta_n$  of categories. For each of the functions *merge* or *move* in a minimalist grammar  $G$  giving a category  $\Delta$  there is a corresponding inference rule. At each parse step the parser applies either *un-merge*

or *un-move* which will then generate a particular derivation tree of this parse step (partial output) and a particular category  $\Delta$ . If *un-merge* is applied, the result will be two branches in the derivation tree with a  $\bullet$  as the root symbol; for *un-move* it will be one branch with a  $\circ$  in root position. The *start* and *scan* rule complement the parser rules. *Start* determines the distinguished category “*c*” to begin with. *Scan* checks if the features currently parsed correspond to the features of a lexical entry. If they do, then the rule on the priority heap is replaced by this lexical entry. *Scan* is applied immediately to obtain an incremental parser.

The following inference rules were originally defined by Main-  
guy (2010, p. 17f.) and are repeated here followed by an example  
parse.

1. “Start rules: for every lexical item  $\gamma :: \delta c$ ,

**Start:**  $\epsilon / \text{start} \longrightarrow [\epsilon / \delta \cdot c]$ ”

The start rule determines the unique symbol “*c*” that initiates the parse. The position index of the start rule is empty (written as  $\epsilon$ ) because this rule is written in root position of the derivation tree.

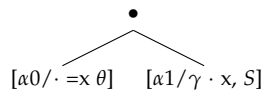
2. “Un-merge rules: the left-hand category is of the form  $[\alpha / \delta = x \cdot \theta, S]$ ”, meaning that the dot is to the right of a selector “=*x*” and the parser is looking for the corresponding category “*x*” to apply *un-merge*.

a) “Cases where the selector was a *simple tree* ( $\delta = \epsilon$ ):

- i. For any lexical item of feature string  $\gamma x$ ,

**Unmerge-1:**

$[\alpha / =x \cdot \theta, S] \longrightarrow$



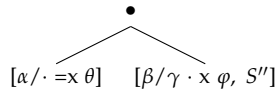
*t* is here a simple category *s* if  $\gamma = \emptyset$  (and thus  $S = \emptyset$  too), and a complex category *c* otherwise.”

*Unmerge-1* is applied when the priority heap contains a rule with a selector “=x” as the leftmost feature of the current rule and no corresponding category “x”. The parser will look for a rule with the category “x” as the rightmost feature in the lexicon and writes this rule to the priority heap. Then, the position indices are updated: 0 is attached to the position indices of the rule with the selector “=x” and 1 to the rule with the category “x”.

- ii. “For any element  $(\gamma \ x \cdot \ \varphi) \in S$ , with  $S'' = S - (\gamma \ x \cdot \ \varphi)$ ,

**Unmerge-3, simple:**

$[\alpha / =x \cdot \theta, \beta / \gamma \ x \cdot \varphi, S''] \rightarrow$



$t$  is here a simple category  $s$  if  $\gamma = \emptyset$  (and thus  $S'' = \emptyset$  too), and a complex category  $c$  otherwise. It should be noted that necessarily,  $\varphi \neq \emptyset$ . ”

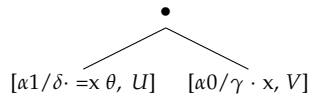
*Unmerge-3, simple* is applied when the priority heap contains a rule with a selector “=x” as the leftmost feature of the current rule and a rule with the corresponding category “x”. No position indices will be added.

- b) “Cases where the selector was a *complex tree*:

- i. For any decomposition  $S = U \sqcup V$ , and any lexical item of feature string  $\gamma \ x$ ,

**Unmerge-2:**

$[\alpha / \delta \cdot =x \cdot \theta, S] \rightarrow$



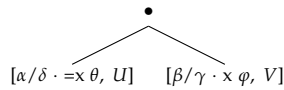
$t$  is here a simple category  $s$  if  $\gamma = \emptyset$  (and thus  $V$  has to be empty too), and a complex category  $c$  otherwise.”

*Unmerge-2* is applied when the current feature is a selector “=x” and a complex tree (meaning that the selector is not the leftmost category of the current rule). The priority heap does not contain a rule with the corresponding category “x” as the rightmost feature and the parser will look for this rule in the lexicon. Then, the position indices are updated: 1 is attached to the position indices of the rule with the selector “=x” and 0 to the rule with the category “x”.

- ii. “For any element  $(\gamma \times \cdot \varphi) \in S$ , and any decomposition  $S = U \sqcup V \sqcup (\gamma \times \cdot \varphi)$ ,

**Unmerge-3, complex:**

$$[\alpha/\delta = x \cdot \theta, \beta/\gamma \times \cdot \varphi, S''] \longrightarrow$$



$t$  is a simple category  $s$  if  $\gamma = \emptyset$  (and thus  $V$  has to be empty too), and a complex category  $c$  otherwise,  $\varphi \neq \emptyset$ .”

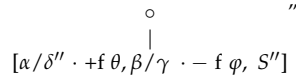
*Unmerge-3, complex* is applied when the current feature is a selector “=x” and a complex tree (meaning that the selector is not the leftmost category of the current rule). The priority heap contains a rule with the corresponding category “x”. No position indices will be added.

- 3. “Un-move rules: the left-hand category is of the form  $[\delta + f \cdot \theta, S]$

- a) For any  $(\gamma - f \cdot \varphi) \in S$  (unique by the SMC), with  $S'' = S - (\gamma - f \cdot \varphi)$ ,

**Unmove-2:**

$$[\alpha/\delta'' + f \cdot \theta, \beta/\gamma - f \cdot \varphi, S''] \longrightarrow$$



*Unmove-2* is applied when the priority heap contains a rule with a licenser “+f” and a (unique) corresponding licensee “-f”. No position indices will be added.

- b) “If there is no  $(\gamma - f \cdot \varphi) \in S$ , then for any lexical item of feature string  $\gamma - f$ ,

**Unmove-1:**

$$[\alpha / \delta'' + f \cdot \theta, S] \longrightarrow \begin{array}{c} \circ \\ | \\ [\alpha 1 / \delta'' \cdot + f \theta, \alpha 0 / \gamma \cdot - f, S] \end{array}$$

*Unmove-1* is applied when the priority heap contains a rule with a licenser “+f” but no rule with the corresponding licensee “-f”. The parser will look for the rule with the licensee “-f” as the rightmost feature in the lexicon. Then, the position indices are updated: 1 is attached to the position indices of the rule with the licenser “+f” and 0 to the rule with the licensee “-f”.

**4.3.3.3 Example Parse**

An example parse of the sentence *The manager will hire the employee* using a similar grammar fragment to the one presented in Section 4.3.1 will show how the parser applies these inference rules. The following grammar fragment is used:

- *the* :: =n d -case
- *manager* :: n
- *will* :: =>v +case t
- *hire* :: =d +case v
- $\epsilon$  :: =>v =d v
- *employee* :: n

The parser begins with the *axiom* of the grammar. The position index of the axiom written before the slash “/” is empty (denoted

by  $\epsilon$ ) because the axiom is always written in root position of the derivation tree.

$$[\epsilon/\epsilon :: =t \cdot c]$$

Now, the parser searches for all possible items in the lexicon whose features end with “t”. There is only one item, *will*, so the parser applies the rule *unmerge-1* and the parser state looks like this:

$$[0/\epsilon :: \cdot =t c], [1/will :: =>v +case \cdot t]$$

The parser obtains the position indices from the *unmerge-1* rule and writes these before “/”. The following position indices are concatenated to the previous ones for each rule separately. For the first rule on the priority heap the dot is in the left-most position, therefore this rule is applicable for the scan operation. Since this yields the distinguished feature “c”, the input will not be shortened and the scan operation succeeds.

Now the parser searches for an entry in the lexicon whose features end with “-case”, which is the corresponding feature for “+case”. There is only one entry, so the *unmove-1* operation is applied:

$$[11/will :: =>v \cdot +case t, 10/the :: =n d \cdot -case]$$

Note that the two rules are written in one pair of square brackets and the rule which triggered the unmove operation is written in first position. Now the parser looks for a lexical entry with “v” as the right-most feature (the complement to “=>v”). There are two possible entries in the lexicon: “*hire* :: =d +case v” and “ $\epsilon$  :: =>v =d v”. The parser uses the oracle as a look-up to determine the correct rule for the next step. In this case the correct rule is “ $\epsilon$  :: =>v =d v” and *unmerge-1* is applied to “*will* :: =>v \cdot +case t”:

$$[10/the :: =n d \cdot -case], [110/will :: \cdot =>v +case t], \\ [111/\epsilon :: =>v =d \cdot v]$$

After a merge operation the rules are written in separate square brackets again. The dot moves to the left-most position for the lexical item *will*. Now the linear order of the constituents on the priority heap changes because the position indices are always in ascending order. This is why the rule for *will* is not applicable for the scan operation yet; it stays on the priority heap and waits for the right moment to be scanned.

In the next step, *unmerge-3 complex* is applicable to “*the :: =n d · -case*” and “*ε :: =>v =d · v*”:

[10/*the :: =n · d -case*], [110/*will :: · =>v +case t*],  
[111/*ε :: =>v · =d v*]

The dot in both rules moves one position to the left. No position indices are added for *unmerge-3 complex*.

Next, the parser looks for a rule in the lexicon with “n” as the right-most feature and finds two possible entries: “*manager :: n*” and “*employee :: n*”. The parser adds both entries separated by a semicolon, indicating alternative lexical entries with equal features. Later, these lexical entries will be checked against the input to determine the matching constituent.

The parser adds “*manager; employee :: n*” directly behind “*the*” to the priority heap by applying *unmerge-1*:

[100/*the :: · =n d -case*], [101/*manager; employee :: · n*],  
[110/*will :: · =>v +case t*], [111/*ε :: =>v · =d v*]

The first three rules on the priority heap can be scanned because their dots are in the left-most position. The scan operation replaces the rules by the corresponding lexical entries and checks against the input which word to choose, *manager* or *employee*. Since the input sentence is *The manager will hire the employee* the parser finds a match for *The manager will*. The parser keeps track of the words already scanned by adding a dot to the input string: *s = The manager will · hire the employee*.

Next, *unmerge-1* is applied to the rule “*ε :: =>v · =d v*”. The parser looks for a rule with “v” as its right-most feature in

the lexicon and finds two rules again, “*hire* :: =d +case v” and “ $\epsilon$  :: =>v =d v”. After consulting the oracle the parser adds “*hire* :: =d +case v” to the priority heap by applying *unmerge-1*:

[1110/ $\epsilon$  :: · =>v =d v], [1111/*hire* :: =d +case · v]

Since the special feature “=>” in the first rule triggers left incorporation, the phonetic material of the lexical item that contains the corresponding feature “v” moves to this rule. Now, the priority heap looks like this:

[1110/*hire* :: · =>v =d v], [1111/ $\epsilon$  :: =d +case · v]

The first rule on the priority heap is applicable for scan and the dot in the input string moves one position to the right:  $s =$  *The manager will hire · the employee*. Next, the feature “+case” triggers *unmove-1* which retrieves the rule for *the* from the lexicon because its features end with “-case”:

[11111/ $\epsilon$  :: =d · +case v, 11110/*the* :: =n d · -case]

Now, the feature “=d” of “ $\epsilon$ :: =d · +case v” and “d” of “*the* :: =n d · -case” trigger *unmerge-3 simple*:

[11110/*the* :: =n · d -case], [11111/ $\epsilon$  :: · =d +case v]

The dot in both rules moves one position to the left and the rules are reordered according to an ascending order of position indices. In the last step, the parser looks for a lexical entry with “n” as its right-most feature and finds the entries for *manager* and *employee* which are added as alternative lexical entries to the priority heap by *unmerge-1*:

[111100/*the* :: · =n d -case], [111101/*manager; employee* :: · n],  
[11111/ $\epsilon$  :: · =d +case v]

The dot is in the left-most position for all items on the priority heap, therefore all are applicable for the scan operation and after



a successful scan the dot in the input string moves to the position behind the last word:  $s = \textit{The manager will hire the employee} \cdot$ . The complete sentence is processed and the priority heap is empty. The parse finishes.

Figure 4.13 shows the derivation tree of the parsing example for the sentence *The manager will hire the employee*.

The rule in root position of the derivation tree has an empty position index because it is the axiom and represents the derived tree itself (Mainguy, 2010). Just below are the two rules obtained by applying unmerge-1. The position indices reflect the position of the relevant categories in the derived tree (0 for going left and 1 for going right from the parent node).

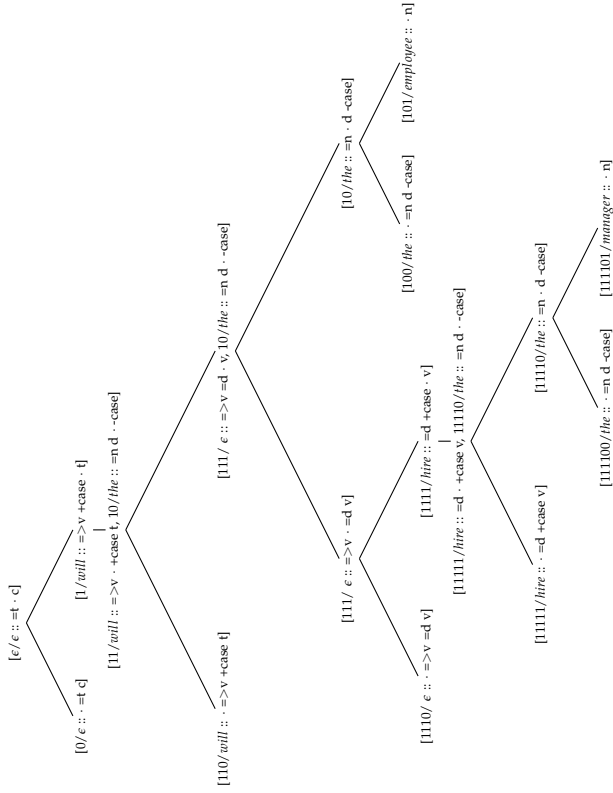


Figure 4.13: Derivation tree for *The manager will hire the employee*

#### **4.3.4 The Complexity Metric**

Parsing a sentence requires the assignment of resources, for instance working memory. Working memory is limited not only in capacity but also in time (Cowan, 2000; Frazier, 1979; Wagers, 2013). Some effort has been made in research on working memory capacity to estimate an upper limit. The most influential number is probably  $7 \pm 2$  proposed by Miller (1956) which is taken to be a limit for short-term memory. The idea of limiting short-term memory in contrast to long-term memory incorporates the notion of time. When understanding language, we need to remember earlier parts of the message until we can integrate these parts into the sentence structure in order to comprehend the whole message.

Rambow and Joshi (1994) proposed a complexity metric based on the grammar formalism TAG which reflects the *amount of time an item is stored in memory by the processor*. The assumption is that the longer a word has to be memorized, the harder it becomes to retrieve the word and integrate it into the current syntactic structure. This idea is similar to working memory decay (Lewis and Vasishth, 2005) and re-integration of long-distance dependencies (Gibson, 1998; Grodner and Gibson, 2005). Kobele et al. (2013) refine this idea and define *stack tenure* which reflects the amount of time a constituent is kept on the stack. The timer for a constituent starts as soon as a minimalist grammar rule predicts the constituent, and its corresponding features are put into the parser state. The timer stops when all features of the constituent are processed and the constituent is scanned.

The complexity metric of this thesis uses *tenure* in a similar way as Kobele et al. (2013). It also accounts for the time an item waits on the priority heap for the right moment to be scanned. In contrast to Kobele et al. (2013) who start the timer as soon as the constituent is predicted by a certain rule, in this thesis the timer for a constituent starts when the features of an item on the priority heap are processed and scan would be applicable. If the constituent is in first position, it can be matched against the input immediately and will be removed from the priority heap. This constituent will

have a tenure value of 0. Sometimes, a constituent has to wait for the right moment to be scanned, for instance when there is a movement to the left; this will add up to the time it stays on the priority heap in terms of parse steps. This way, the complexity metric is accounting for the structural complexity of the intervening structure by considering the number of grammar operations the parser has to apply in between.

Tenure is considered to be an online measure calculated for each word and reflects processing difficulty at particular points in the sentence (similar to predictions from DLT (Gibson, 1998)). The offline measure of the complexity metric used in this thesis is *average tenure* which is the sum of tenure values for each word divided by the number of words in the sentence. Average tenure predicts overall acceptability or unacceptability of sentences (see Section 5 for details on the results).<sup>4</sup>

The derivation tree in Figure 4.14 illustrates tenure values for the sentence *The manager will hire the employee*. The structure of the tree is the same as the structure of the derivation tree in Figure 4.13. In Figure 4.14, each node represents either a merge or a move operation. The number before the colon indicates the number of the parse step. The axiom is put onto the priority heap in step 0 and the first merge operation is *unmerge-1*. Following the notation of Kobele et al. (2013), each lexical item is superscripted on the left with a number indicating the parse step at which this item is scanned and put onto the priority heap. The superscript on the right gives the number of the parse step at which this item is removed from the priority heap.

---

<sup>4</sup>Kobele et al. (2013) used *maximum tenure*, i.e., the maximum number of nodes in the derivation, as an offline measure for the complexity of a sentence. Graf and Marcinek (2014) complement this measure with other offline metrics: the maximum tenure of all leaves in the derivation, the maximum number of nodes with tenure strictly greater than 2 and the maximum number of leaves with tenure strictly greater than 2.

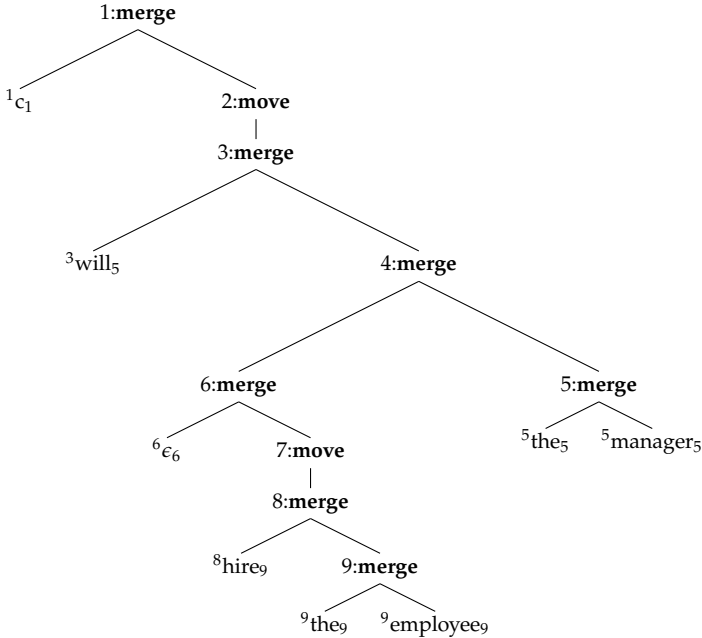


Figure 4.14: Derivation tree with tenure counts for *The manager will hire the employee*

The calculated tenure values for each word are given below:

The manager will hire the employee.  
 0 0 2 1 0 0

As Kobele et al. (2013) noted, the crucial point of the minimal analysis is that the inflectional head (*will*) is predicted quite early during the parse and is merged with the verb (*hire*) only after all its arguments have been merged with it. Therefore the inflectional head stays on the priority heap for a long time and receives a high tenure value.

Average tenure value is calculated by dividing the sum of the tenure values by the number of words which yields an average

tenure of  $\frac{1}{3}$  for the given sentence. This example is simply an illustration of how the calculation works. The values for tenure and average tenure are only meaningful when compared to a different sentence, which will be done in the results section of this thesis (Section 5).

## 5 Results

This chapter will present the results calculated by the algorithm described in Section 4. In particular we will compare the predictions of the offline and online complexity metric to a well-established theory of sentence processing difficulty, the Dependency Locality Theory (DLT, (Gibson, 1998, 2000)).

As described in Section 4.3.4, the complexity metric *tenure* reflects the amount of time an item is stored in memory by the parser, and the algorithm (described in Section 4.3) calculates tenure values for each word in the sentence. The offline measure *average tenure* is the sum of all tenure values of all words in the sentence divided by the number of words in the sentence. Hence, average tenure yields one value for the complete syntactic structure of the sentence and can predict acceptability ratings for sentences. We use average tenure instead of the maximum tenure value of the sentence, because average tenure was a better predictor of the data in this thesis. Table 5.1 shows the measures that are compared to each other in this chapter. DLT is based on Gibson (2000) and sums up integration and storage cost. To obtain a value for the whole sentence, similar to average tenure, *maxDLT* is calculated, which gives the maximum DLT value of the sentence. We decided to use the maximum value of DLT here, because Gibson (1998, p.16) states “it is assumed that the relative intuitive acceptability of two unambiguous sentences which are controlled for plausibility is determined by the maximal quantity of memory resources that are required at any point during the parse of each sentence.”. For sake of completeness we calculated the average DLT values, but these values make the same predictions as maximum DLT, so we will not report average DLT values in this thesis.

**Table 5.1:** Comparison of data and predictions of the two complexity metrics

Data	DLT (Gibson, 2000)	Tenure
Acceptability ratings	max Integration +Storage cost	Average tenure
Reading times at the current word	Integration+Storage cost at the current word	Tenure at the current word

We include tests for *goodness of fit* between the online data, i.e., the reading times of the self-paced reading experiments discussed in Section 5.2, and the predicted values from DLT and tenure. We calculate a simple linear regression and use the  $R^2$  value to indicate how well the two complexity metrics predict the data. In a simple linear regression,  $R^2$  is the square of the correlation between the observed outcomes and the predicted values. The higher the value of  $R^2$  the better the model fit.

Following the approach of Rambow and Joshi (1994) we will only use unambiguous structures to determine the complexity of the syntactic structure that might lead the parser to break-down. The results chapter is structured as follows. Firstly the two offline metrics, maxDLT and average tenure, are evaluated using offline data such as acceptability ratings and comprehensibility judgments in Section 5.1. The phenomena described in this section of the chapter are nesting constructions (Cowper, 1976; Gibson and Thomas, 1997), the asymmetry between center-embedded and right-branching structures (Miller and Chomsky, 1963) and subject-modified versus object-modified subject-extracted (SRC) or object-extracted (ORC) relative clauses (Gibson and Thomas, 1996, 1997). In the second part of the chapter, Section 5.2, word-by-word predictions for online self-paced reading data are derived from the online measures DLT and tenure. The predictions cover SRC versus ORC (Grodner and Gibson, 2005), data for sentences inducing locality effects in English (Grodner and Gibson, 2005), and structures with an intervening complex noun phrase including a prepositional phrase versus an intervening relative clause (Gibson and Warren, 2004).



## 5.1 Modeling the Offline Data

### 5.1.1 Nesting Constructions: SC/RC vs. RC/SC Embedding

Cowper (1976) observed a contrast in different types of *nesting constructions*. In particular she tested if a relative clause inside a sentential subject (SC/RC) such as sentence (62) or a sentential subject inside a relative clause (RC/SC) like example (61) causes more processing difficulty.

(61) Relative clause, then sentential complement (RC/SC)

# The executive [who the fact [that the employee stole office supplies] worried] hired the manager.

(62) Sentential complement, then relative clause (SC/RC)

The fact [that the employee [who the manager hired] stole office supplies] worried the executive.

Results from acceptability judgment experiments (Gibson and Thomas, 1996, 1997) showed that constructions like (62) are acceptable while sentences like (61) are unacceptable (Cowper, 1976; Gibson, 1991) (see Table 5.2 for details).

The complexity metrics proposed by Abney and Johnson (1991); Bever (1970); Miller and Chomsky (1963); Gibson (1991); Kimball (1973); Lewis (1993); Stabler (1994) cannot account for this contrast in processing difficulty (Babyonyshev and Gibson, 1999). For instance, the principle of two incomplete sentences from Kimball (1973) would predict that both examples lead to a breakdown of the parser because at the most embedded subject *the manager* in (62) and *the employee* in (61) there are three incomplete sentences (Gibson, 1998).

DLT as proposed by Gibson (2000) can account for this processing asymmetry, because it accounts for the fact that there is only one long incomplete dependency – the SC verb – in (62) while in (61) there are two long incomplete dependencies – the RC verb and the RC empty category position (Babyonyshev and Gibson, 1999; Gibson and Thomas, 1997).

**Table 5.2:** Comparison of data (standard errors in parentheses) and predictions of maxDLT and average tenure for nesting constructions SC/RC vs. RC/SC

Structure	Acceptability rating	MaxDLT	Average tenure
#RC/SC (61)	3.38 (.08)	9	3.57
SC/RC (62)	1.92 (.08)	6	3.5

Table 5.2 shows the results for the comparison of the acceptability ratings and the offline predictions based on maxDLT and average tenure. Gibson and Thomas (1997, p. 22) used an acceptability rating experiment with a scale ranging from 1 (best) to 5 (worst), thus the higher values in the table represent lower acceptability. The unacceptable construction is marked with a hash sign # from now on.

Both complexity metrics predict the correct asymmetry for these examples in that sentence (61) is more difficult to comprehend than sentence (62). The maximum value predicted by DLT is 9 in (61) and 6 in (62) for the verb *worried* (Gibson, 2000). Tenure predicts the same point of highest processing difficulty as DLT (a maximum tenure value of 22 in (62) and 25 in (61) for the verb *worried*). Nevertheless, we compare here the maximum DLT value to the average tenure value. The difference in values for average tenure is not as pronounced as the difference in acceptability ratings. Interestingly, average tenure is able to capture the difference between the two syntactic constructions even though both sentences contain the same syntactic structures only in a different order. Hence, the presented algorithm is able to capture these subtle differences in the overall syntactic structure.

### 5.1.2 Center-embedded vs. Right-branching Structures

An extensively studied phenomenon in the psycholinguistic literature is *center-embedded structures*, whose difficulty increases with the number of embeddings until the sentence becomes unacceptable (Cowper, 1976; Frazier and Fodor, 1978; Gibson and

Thomas, 1999; Kimball, 1973; Lewis, 1993; Miller and Chomsky, 1963; Stabler, 1994; Yngve, 1960, among many others). Chomsky (1965) states that these constructions are *unacceptable* but *grammatical*. Most psychologists and linguists have assumed that there must be a psychological rather than a linguistic explanation for the difficulty of center-embedded sentences (e.g., Miller and Chomsky, 1963).

Cowper (1976) uses the following examples to exemplify the contrast between center-embedded and right-branching structures.

- (63) Center-embedded structure, 1 clause

The executive [who the manager worried] hired the employee.

- (64) Right-branching structure, 1 clause

The manager worried the executive [who hired the employee].

- (65) Center-embedded structure, 2 clauses

#The executive [who the manager [who the woman saw] worried] hired the employee.

- (66) Right-branching structure, 2 clauses

The woman saw the manager [who worried the executive] [who hired the employee].

In sentence (63) the relative clause *who the manager worried* is embedded in the matrix sentence *the executive ... hired the employee*. In example (65) a second relative clause *who the woman saw* is embedded in the first relative clause, resulting in a sentence that is extremely hard to process and which is therefore perceived as unacceptable (Cowper, 1976).

The right-branching versions of the center-embedded structures cause no processing difficulty even with deep embedding. According to Gibson (1991) (see also Gibson (1998)) the processing difficulty is language independent. The unacceptability of center-embedded structures is attributed to the limits of the

**Table 5.3:** Comparison of data and predictions of maxDLT and average tenure for center-embedded vs. right-branching structures

Structure	Comprehensibility judgment	MaxDLT	Average tenure
CE 1 clause (63)	96	3	2.67
RB 1 clause (64)	97	3	1.22
CE 2 clauses (65)	69	7	5
RB 2 clauses (66)	96	3	1.31

computational resources of the language processor (Abney and Johnson, 1991; Chomsky, 1965; Kimball, 1973; Lewis, 1993; Rambow and Joshi, 1994; Stabler, 1994).

Table 5.3 compares the results of a comprehensibility judgment experiment by Hamilton and Deese (1971) with the predictions of maxDLT and average tenure. In the experiment conducted by Hamilton and Deese (1971), participants judged the comprehensibility of sentences with an increasing number of embedded clauses in right-branching and center-embedded structures. Participants listened to the sentence and had to indicate whether the sentence was comprehensible or incomprehensible in a two-second interval following each sentence. Table 5.3 gives the percentage of sentences judged as comprehensible for each sentence type (center-embedded vs. right-branching) and length (one clause vs. two clauses). The right-branching sentences were judged as most comprehensible and the two-level center-embedded structures as least comprehensible. The comprehensibility of right-branching structures was not affected by length, whereas the comprehensibility of the center-embedded sentences was affected (69 % for two levels of embedding vs. 96 % for one level of embedding). These results are in line with the proposal that the deeper embedding of right-branching structures does not affect processing.

Both complexity metrics, maxDLT and average tenure, correctly predict sentence (65) to be the least comprehensible structure. Interestingly, maxDLT predicts the same maximum DLT value of 3 for the other three constructions. However, average tenure pre-

**Table 5.4:** Comparison of data and predictions of DLT and tenure for the embedded verbs of the two-level center-embedded sentence

Sentence	Verb	DLT	Tenure
CE 2 clauses (65)	saw	3	19
CE 2 clauses (65)	worried	7	19
CE 2 clauses (65)	hired	5	27

dicts more graded results with, sentence (65) being the hardest structure followed by the one-level center-embedded construction (63), the two-level right-branching structure (66) and the one-level right-branching example (64). This more graded prediction is expected since tenure takes into account the syntactic structure of the given constructions and accounts for the time an item has to be stored in memory, which explains the slight length effects.

One interesting result for these constructions is the prediction of the position of highest processing difficulty in the two-level center-embedded sentence (65). Table 5.4 shows the predictions by DLT and tenure for the three embedded verbs in (65). DLT predicts the highest processing difficulty at the second verb *worried*, while tenure predicts the highest processing difficulty at the third verb *hired*. Vasishth et al. (2010) conducted a self-paced reading and an eye-tracking experiment with center-embedded structures in English and found a non-significant trend for longer reading times at the third verb in the self-paced reading experiment (see Figure 2, p. 543, Vasishth et al. (2010)) and longer re-reading times at the third verb in the eye-tracking experiment (see Figure 4, p. 549, Vasishth et al. (2010)). Even though this trend for longer reading times on the last verb compared to the second one was not significant, it reflects longer processing at the last verb, which is predicted by tenure but not by DLT. Tenure accounts for the longer storage of the first NP which needs to be linked syntactically to the last VP, hence a length effect due to more intervening structure is predicted and could be the reason for slightly longer reading times on the last verb compared to the second.

### 5.1.3 Subject-modified vs. Object-modified SRC or ORC

The previous section discussed center-embedded sentences that were embedded in object position of the matrix sentence. To test whether the position of the embedded relative clauses makes a difference for the observed processing difficulty, Gibson and Thomas (1997) tested various types of constructions for which relative clauses were embedded inside relative clauses either in subject or object position of the matrix clause. They used the following constructions: object-extracted relative clause embedded in another object-extracted relative clause in matrix subject position as in (67); object-extracted relative clause embedded in a subject-extracted relative clause in matrix subject position as in (68); object-extracted relative clause embedded in another object-extracted relative clause in matrix object position as in (69); and an object-extracted relative clause embedded in a subject-extracted relative clause in matrix object position as in (70). The last sentence (71) is the corresponding right-branching structure.

- (67) Object-extracted, object-extracted in matrix subject position  
The parishioners [who the actress [who the preacher was sleeping with] had scandalized] were overlooked by the newsman.
- (68) Object-extracted, subject-extracted in matrix subject position  
The parishioners [who the actress [who was sleeping with the preacher] had scandalized] were overlooked by the newsman.
- (69) Object-extracted, object-extracted in matrix object position  
The newsman overlooked the parishioners [who the actress [who the preacher was sleeping with] had scandalized].
- (70) Object-extracted, subject-extracted in matrix object position  
The newsman overlooked the parishioners [who the actress [who was sleeping with the preacher] had scandalized].

## (71) Right-branching

The preacher was sleeping with the actress [who had scandalized the parishioners] [who were overlooked by the newsman].

In English an object-extracted relative clause is predicted to be harder to process than a subject-extracted relative clause (Gibson and Thomas, 1997; Gibson and Fedorenko, 2013; Just and Carpenter, 1992; King and Just, 1991, among others). SPLT, as discussed by Gibson and Thomas (1997), predicts that the position of embedding in the matrix clause will not make a difference for the acceptability rating results because there is no extra memory cost associated with maintaining the top-level clause predictions.

Gibson and Thomas (1997) conducted an acceptability rating experiment. Participants had to rate the acceptability of the presented sentences on a scale from 1 (best) to 5 (worst), judging how well they understood the sentences. The higher the rating the lower the acceptability. The results for the four constructions tested by Gibson and Thomas (1997) are given in Table 5.5.<sup>1</sup>

**Table 5.5:** Comparison of data (standard errors in parentheses) and predictions of maxDLT and average tenure for subject-modified vs. object-modified SRC or ORC

Structure	Acceptability rating	MaxDLT	Average tenure
(67)	3.02 (.11)	7	9.22
(68)	2.52 (.11)	7	8.22
(69)	2.79 (.11)	3	7.06
(70)	2.64 (.11)	3	5.94
(71)	-	3	1.5

Their results show no significant effect of extraction-type (object-extracted vs. subject-extracted sentences), no effect for the position of the embedding (matrix subject or matrix object position) and no

<sup>1</sup>Gibson and Thomas (1997) did not include the right-branching structure in their experiment. However, we include this structure in the table to highlight the difference in predictions by maxDLT and average tenure.

interaction. Since these results are not statistically reliable we cannot draw strong conclusions from them. Further, an experiment with all five constructions is needed to directly compare all syntactic structures. Nevertheless, we discuss here predictions made by maxDLT and average tenure. DLT predicts the lowest acceptance rate for the first two sentences (67) and (68). Both are relative clauses that are embedded in subject position of the matrix clause. Importantly, maxDLT does not predict that object-extracted relative clauses are less acceptable than their subject-extracted counterparts. In contrast, average tenure predicts that the object-extracted examples should be harder to process than the subject-extracted structures, which is in line with previous findings in the literature. Further, the relative clauses embedded in matrix subject position should be more difficult to process than relative clauses embedded in object position of the matrix clause according to the predictions of average tenure. Note that maxDLT predicts the same value for the right-branching structure as for two of the center-embedded sentences ((69) and (70)). Average tenure assigns a fairly low value to the right-branching structure because it is undoubtedly the easiest construction among the five examples.

Interestingly, the online complexity metric tenure predicts the highest processing difficulty at the auxiliaries in the sentence. So far, only one study by Warren and Gibson (2002) included auxiliaries in an experiment testing predictions of DLT. However, they combine the auxiliary and the verb into one verbal phrase and do not give word-by-word predictions. Hence, the predictions by DLT for an auxiliary are not clear.

## **5.2 Modeling the Online Data**

### **5.2.1 Subject-extracted vs. Object-extracted Relative Clauses**

Previous work has established that, in English, subject-extracted relative clauses (SRC) are easier to process than object-extracted relative clauses (ORC) (Gordon et al., 2001; Grodner and Gib-



son, 2005; Just and Carpenter, 1992; King and Just, 1991; Wanner and Maratsos, 1978). Memory-based approaches assume that the parser favors the most recent attachment site of the syntactic structure to prevent long-distance dependencies (Frazier, 1979; Grodner and Gibson, 2005). Further, Gibson and Warren (2004) discuss two possible explanations as to why the integration of the verb *sent* in the ORC (73) should be more difficult than in the SRC (72): in the ORC (1) the noun phrase *the photographer* is integrated into the syntactic structure as the subject of *sent*, and (2) *who* is integrated as the object of *sent*. In contrast there is only one integration at *sent*, in the SRC (*who* is integrated as the subject of *sent*). The difference between two integrations for the ORC compared to one integration for the SRC at the embedded verb *sent* leads to a slowdown in reading times at *sent* in (73) compared to (72) (Gibson, 2000; Grodner and Gibson, 2005).

SRCs and ORCs are an interesting phenomena to test the predictions of memory-based theories because they differ only in the position of two words, i.e., the embedded noun phrase and the verb. Grodner and Gibson (2005) conducted a self-paced reading experiment with sentences (72) and (73) to test the predictions of the memory-based approach DLT.

(72) Subject-extracted relative clause (SRC)

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

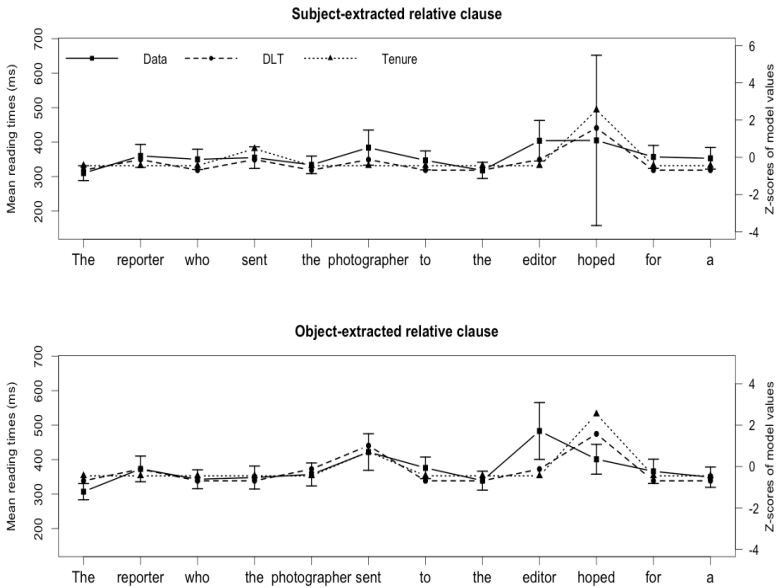
(73) Object-extracted relative clause (ORC)

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

Table 5.6 and Figure 5.1 show the mean reading times of the self-paced reading experiment by Grodner and Gibson (2005) and the word-by-word predictions of the two online complexity metrics DLT and tenure<sup>2</sup>. The detailed results for the main and embedded verb are given below in Table 5.7.

<sup>2</sup>Unfortunately, Grodner and Gibson (2005) only give the mean reading times until the word *i* in the sentence.





**Figure 5.1:** Comparison of data (with 95 % approximate confidence intervals) from Grodner and Gibson (2005) and the models' predictions for the SRC and the ORC

Figure 5.1 shows the results of the comparison between the mean reading times and the predictions by DLT and tenure for each word. The values of DLT and tenure are z-scored so that both complexity metrics can be plotted on the same scale in the graph. Overall, the patterns predicted by DLT and tenure are similar. DLT fits the data for both sentences a bit better ( $R^2=0.357$ ) than tenure ( $R^2=0.142$ ). The divergence between the models and the data could be due to the fact that there is a peak at the noun *editor* in both conditions and at the noun *photographer* in the SRC which is not predicted by either complexity metric. Both metrics predict the highest processing difficulty on the verbs rather than the NPs.

**Table 5.7:** Comparison of data (standard errors in parentheses) and predictions by DLT and tenure for the embedded verbs, data from Experiment 1, Grodner and Gibson (2005)

Sentence	Verb	Reading time	DLT	Tenure
SRC (72)	sent	355 (16)	1	7
SRC (72)	hoped	405 (126)	4	23
ORC (73)	sent	422 (27)	3	9
ORC (73)	hoped	401 (22)	4	23

The critical regions in the sentence are the embedded verb *sent* and the matrix verb *hoped*. Grodner and Gibson (2005) found a significant main effect of extraction-type at the embedded verb *sent* with longer reading times in ORCs compared to SRCs. DLT predicts these results, namely a higher processing cost for the embedded verb *sent* in the object-extracted relative clause compared to the subject-extracted relative clause, because in the ORC the verb *sent* must first be linked to its subject which crosses one discourse referent. Then the object of *sent* has to be integrated with *who* which is co-indexed with *the reporter*. Therefore, DLT yields a value of 3 for *sent* in the ORC compared to 1 in the SRC. According to DLT there should be no processing difference between the conditions at the matrix verb *hoped*. Tenure predicts the same direction of results as DLT for these constructions because objects are more deeply embedded than subjects, hence the movement of a subject causes fewer nodes to be delayed in processing than the movement of an object.

### 5.2.2 Locality Effects

A locality effect occurs when increasing the distance between a dependent and its head results in increased processing difficulty. Grodner and Gibson (2005) tested the locality hypothesis in English in a self-paced reading experiment. Vasisht and Drenhaus (2011) investigated the locality effect in German using self-paced reading, eye-tracking and event-related potentials (ERP). Their findings corroborate the results of Grodner and Gibson. When

Vasishth and Drenhaus (2011) increase the distance between the argument and the verb (critical region) the results show an increase in reading times on the critical and the post-critical region in self-paced reading and second-pass measures for eye-tracking. In the ERP study Vasishth and Drenhaus (2011) observed a negative-going potential around 300–500ms from the onset of the verb which they interpret as reflecting the retrieval process. Bartek et al. (2011) present four experiments to replicate the results of Grodner and Gibson (2005) using self-paced reading and eye-tracking. Their results show that locality effects can be found in relatively difficult and relatively simple structures alike. Similar to the findings of Vasishth and Drenhaus (2011), the observed self-paced reading results correspond to patterns found in later eye-tracking measures (rereading and regression measures).<sup>3</sup>

Next, we will discuss the details of Grodner and Gibson’s original experiment, their results and the word-by-word predictions of the online complexity metrics DLT and tenure.

(74) Matrix – unmodified subject

The nurse supervised the administrator while ...

(75) Matrix – PP-modified subject

The nurse from the clinic supervised the administrator while ...

(76) Matrix – RC-modified subject

The nurse who was from the clinic supervised the administrator while ...

(77) Embedded – unmodified subject

The administrator who the nurse supervised scolded the medic while ...

---

<sup>3</sup>There have been numerous studies (Konieczny, 2000; Levy and Keller, 2013; Levy et al., 2013; Vasishth and Lewis, 2006, among others) that found an anti-locality effect, e.g. a speed-up in reading times when increasing the distance between a dependent and its head. However, we will focus on the studies that found a locality effect.

(78) Embedded – PP-modified subject

The administrator who the nurse from the clinic supervised scolded the medic while ...

(79) Embedded – RC-modified subject

The administrator who the nurse who was from the clinic supervised scolded the medic while ...

Grodner and Gibson (2005) used the six conditions given in (74)–(79). They increased the amount of material between the embedded subject *the nurse* and the embedded verb *supervised*. The additional material modified either the matrix subject ((75)–(76)) or the embedded subject ((78)–(79)). In particular, in (74) and (77) no material was added, in (75) and (78) a prepositional phrase (PP) of three words was added, and in (76) and (79) a relative clause (RC) of five words was added. Grodner and Gibson (2005) focused on the main and embedded verbs as primary regions of interest.

The mean reading times at each word in the sentence are given in Table 5.8 for (74)–(76) and in Table 5.10 for (77)–(79). The results of Grodner and Gibson (2005) show an monotonic increase of reading times at the main and embedded verbs from (74) to (76) and (77) to (79), as is expected by memory-based accounts. Establishing a dependency between the subject and the verb takes longer when there is more distance between them.



**Table 5.9:** Comparison of data (standard errors in parentheses) and predictions of DLT and tenure for the embedded verb *supervised*; data from Experiment 2, Grodner and Gibson (2005)

Sentence	Verb	Reading time	DLT	Tenure
unmodified (74)	supervised	375 (15.2)	1	5
PP-modified (75)	supervised	393 (19.6)	2	10
RC-modified (76)	supervised	389 (17.6)	3	18

Table 5.8 presents the word-by-word predictions of DLT and tenure for the sentences (74)–(76). In line with the results from Grodner and Gibson (2005), DLT and tenure predict a locality effect on the main verb *supervised* for these constructions. With increasing intervening material (additional discourse referents or additional intervening structure) between the noun phrase *the nurse* and the verb *supervised* the predicted values of DLT and tenure increase. A difference between the predictions of the two complexity metrics and the mean reading times occurs for (75) in which a prepositional phrase modifies the subject *the nurse*. Here, the mean reading times are numerically longer for *supervised* compared to (76) which has an intervening relative-clause. This is not expected by the locality hypothesis because the relative clause follows a prepositional phrase and introduces five more words (compared to two additional words for the prepositional phrase). However, this difference between the reading times on *supervised* in (75) and (76) was not statistically reliable and can only be considered as a numerical trend.



Table 5.10: Word-by-word predictions for data from Experiment 2, Grodner and Gibson (2005)

Measure	Input Word	The	admin.	who	the	nurse	supervised	scolded	the	clinic	the	medic
(77)	The	323	389	396	345	373	448	500	439	386		
Reading time	DLT	0	1	0	0	1	3	3	0	1		
Tenure		0	0	0	0	0	8	16	0	0		
(78)	The		admin.	who	the	nurse	from	the	clinic			
Reading time	DLT	322	362	339	349	350	377	313	361			
Tenure		0	1	0	0	1	0	0	1			
		0	0	0	0	0	0	0	0			
	supervised		scolded	the	medic							
Reading time	DLT	466	558	432	385							
Tenure		5	4	0	1							
		18	21	0	0							
(79)	The		admin.	who	the	nurse	who	was	from	the		
Reading time	DLT	322	414	352	356	404	370	459	366	319		
Tenure		0	1	0	0	1	0	1	0	0		
		0	0	0	0	0	0	27	8	0		
	clinic		supervised	scolded	the	medic						
Reading time	DLT	352	588	538	438	400						
Tenure		1	7	5	0	1						
		0	8	29	0	0						

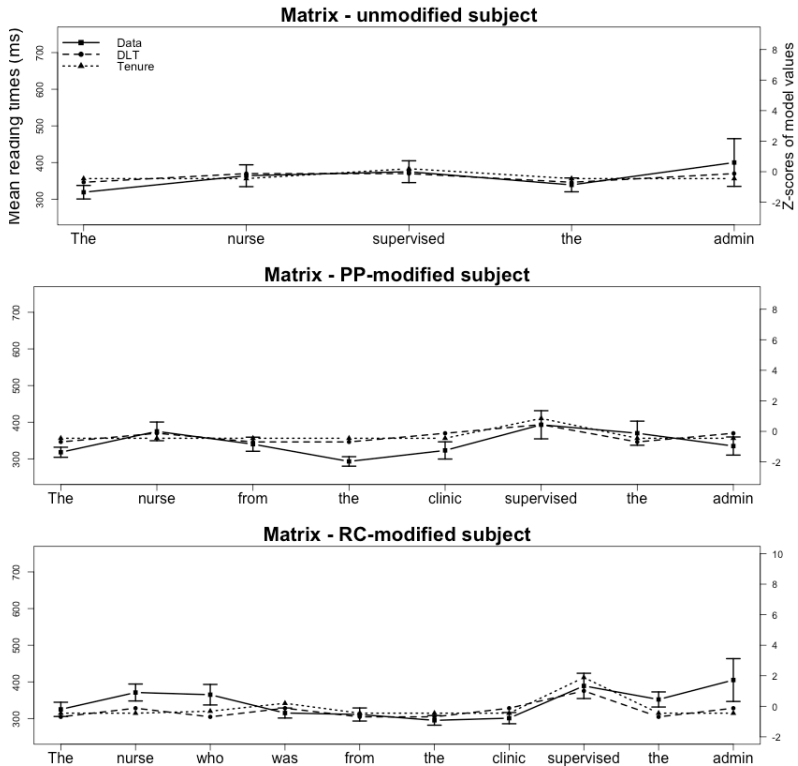
**Table 5.11:** Comparison of data (standard errors in parentheses) and predictions of DLT and tenure for the embedded verbs *supervised* and *scolded*; data from Experiment 2, Grodner and Gibson (2005)

Sentence	Verb	Reading time	DLT	Tenure
unmodified (77)	supervised	375 (15.2)	1	8
PP-modified (78)	supervised	393 (19.6)	2	18
RC-modified (79)	supervised	389 (17.6)	3	8
unmodified (77)	scolded	500 (39.7)	3	16
PP-modified (78)	scolded	558 (43.8)	4	21
RC-modified (79)	scolded	538 (50.3)	5	29

Table 5.10 shows the mean reading times for each word measured by Grodner and Gibson (2005) and the word-by-word predictions of DLT and tenure for (77)–(79). The results of Grodner and Gibson (2005) show the highest mean reading times in all three sentences for the main and embedded verbs, confirming a locality effect for these constructions. Table 5.11 gives the mean reading times and predictions of DLT and tenure for the verbs as the primary region of interest. The reading times on the embedded verb *supervised* and the main verb *scolded* show a similar pattern as in (74)–(76). In the construction with the prepositional phrase *supervised* and *scolded* were read numerically longer compared to the construction with the intervening relative clause and the construction without any intervening material. Again, this difference was not statistically reliable for either of the verbs.

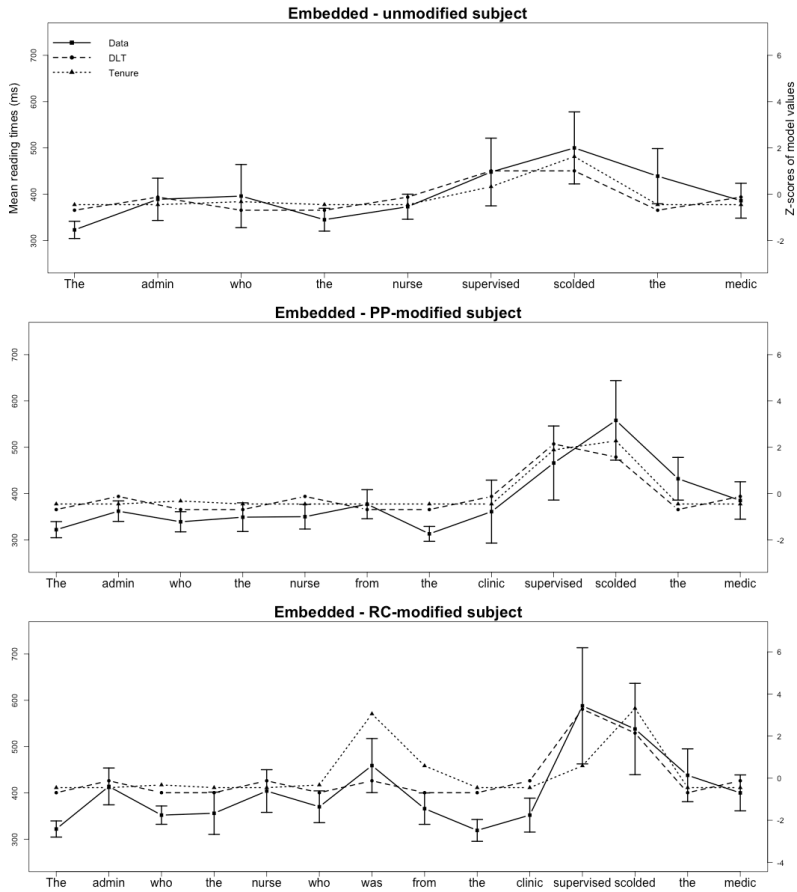
DLT predicts a linear increase in complexity for both verbs from (77) to (79). The values predicted by tenure show a slightly different pattern. Similar to DLT, tenure predicts an increase of complexity for the main verb *scolded*, but the highest complexity for the embedded verb *supervised* is predicted for (78) compared to an equal complexity value for *supervised* in the other two constructions (77) and (79). Tenure predicts a lower value for *supervised* in (79) because the intervening structure for the prepositional phrase *from the clinic* is already resolved at this point in the sentence. In particular, the intermediate structure of the prepositional phrase is already attached to the verb *was* inside the relative clause,

therefore it is not stored in memory anymore but integrated into the overall structure of the sentence. Hence, tenure predicts a lower structural complexity for the embedded verb *supervised* in (79) compared to (78).



**Figure 5.2:** Comparison of data (with 95 % approximate confidence intervals) for Expt. 2 (matrix conditions) by Grodner and Gibson (2005) and the models' predictions for the locality examples

Figure 5.2 and Figure 5.3 show the mean reading times and the word-by-word predictions for all conditions (74)–(79). The fit between DLT and the data yielded an  $R^2 = 0.614$  and for tenure an  $R^2 = 0.442$ . Even though the overall fit of DLT to the data is better



**Figure 5.3:** Comparison of data (with 95 % approximate confidence intervals) for Expt. 2 (embedded conditions) by Grodner and Gibson (2005) and the models' predictions for the locality examples

than the fit for tenure, it is worth looking at the  $R^2$  values for each of the sentences in more detail because DLT and tenure differ in their respective fits to the data. Table 5.12 shows the R-squared values for DLT and tenure. Tenure has a better fit to the data than DLT for (75), (77) and (78). Interestingly, (75) and (78) are the structures in which the subject is modified by the prepositional phrase. For these structures tenure predicted a higher value than DLT for the verbs. The better model fit suggests that tenure is better able to handle these constructions. However, there are structures for which DLT explains more of the data. In particular, the longer sentences (76) and (79) yield a higher adjusted R-squared value for DLT compared to tenure. The graphs for these constructions in Figure 5.2 and Figure 5.3 seem to suggest that tenure predicts values that are too big here. The line for tenure is mostly above the line for DLT. This difference between the predictions of tenure and DLT needs to be investigated further.

**Table 5.12:** Comparison of data fit for DLT and tenure

Sentence	$R^2$ DLT	$R^2$ tenure
Matrix – unmodified (74)	0.778	0.077
Matrix – PP-modified (75)	0.343	0.359
Matrix – RC-modified (76)	0.230	0.129
Embedded – unmodified (77)	0.540	0.637
Embedded – PP-modified (78)	0.629	0.756
Embedded – RC-modified (79)	0.803	0.382

When comparing the reading times at the verbs of the unembedded and embedded versions, the verbs in the embedded versions yield a stronger locality effect than the verbs in the unembedded sentences. Grodner and Gibson (2005) argue that this could be due to the filler-gap dependency contained in the embedded versions, while there is no such dependency in the unembedded conditions. When comparing the differences in reading times for the matrix verbs (*supervised* in (74) to (76) and *scolded* in (77) to (79)) it is interesting that DLT and tenure also predict a higher complexity for

the matrix verb in the embedded as opposed to the unembedded versions. However, if these higher complexity values are due to linear-based or the structural-based distance is not clear at this point. It certainly shows that both complexity measures are sensitive to an open filler-gap dependency in the sentence.

### 5.2.3 Complex NP Including a PP Modifier vs. RC Intervening

In the last part of the results chapter we will discuss results for two structures that are matched for length of the intervening material, i.e., the constructions have the same number of discourse referents intervening between the filler and the gap. Both structures contain a long-distance extraction of a *wh*-filler. The difference between the two constructions is the syntactic structure of the intervening material. Gibson and Warren (2004) used the materials given in (80) and (81) below in a self-paced reading experiment.

- (80) The manager<sub>*i*</sub> who<sub>*i*</sub> **the consultant claimed** *t<sub>*i*</sub>* **that** the new proposal had pleased *t<sub>*i*</sub>* will hire five workers tomorrow.
- (81) The manager<sub>*i*</sub> who<sub>*i*</sub> **the consultant's claim about** the new proposal had pleased *t<sub>*i*</sub>* will hire five workers tomorrow.

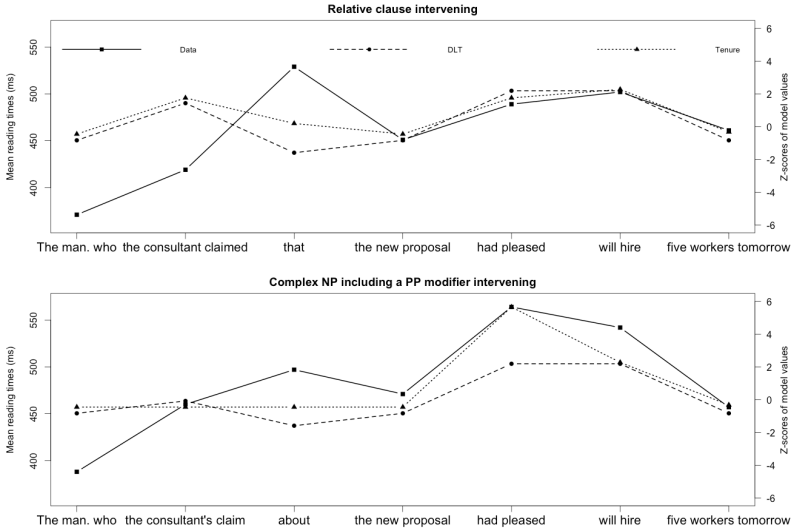
Syntactic theories agree that there is a co-indexed trace *t<sub>*i*</sub>* that can represent a placeholder for intermediate syntactic structure. In example (80) the filler *who* is co-indexed with the subject *the manager* and there are two traces, one after *claimed* representing *the manager* being the object of *claimed*, and one after *had pleased* which can be interpreted as the clause *the new proposal had pleased the manager*. The intermediate trace after *claimed* is necessary because a phrase cannot cross more than one bounding node (NP or IP) at each movement step, otherwise it would violate the grammatical principle called *Subjacency* (Chomsky, 1973). Sentence (81) contains only one trace after *had pleased* because the phrase *the consultant claimed* is nominalized to *the consultant's claim* in this example. Further, there is no intermediate structure with a trace between *who* and the object position of the verb *pleased* in (81).

There is one confound in the materials, namely that the distance between *pleased* and the head noun *proposal* is shorter in (80) than the distance between *pleased* and the head noun *claim* in (81). To rule out this confound Gibson and Warren (2004) added two non-extracted control conditions with the same subject-verb distance at *pleased* to their experiment. The results from these controls show that the readings times of the extracted conditions (80) and (81) were not influenced by this confound in the materials.

Table 5.13 presents the mean reading times for regions of interest defined by Gibson and Warren (2004) and the predictions of tenure and DLT for these regions.







**Figure 5.4:** Comparison of data by Gibson and Warren (2004) and the predictions of tenure and DLT for examples containing an intervening complex NP and PP vs. a RC

The critical region is the VP *had pleased*. The experimental results of the self-paced reading experiment show that participants read the VP *had pleased* significantly slower in (81) compared to (80). However, this effect was only present in the extracted conditions and not in the non-extracted control conditions. Hence, it might only be a reflection of the significant interaction of extraction type and type of intervening phrase that was also found by Gibson and Warren (2004).

Figure 5.4 presents the mean reading times and the predictions by DLT and tenure for the regions defined by Gibson and Warren (2004). The fit between DLT and the data yielded an  $R^2 = 0.168$  and for tenure an  $R^2 = 0.384$ . For these examples tenure fits the data better than DLT as is expected since DLT predicts no processing difference for the two sentences, but tenure does. At the beginning there is a divergence between the data and the values predicted by

**Table 5.14:** Comparison of data and predictions by tenure for the verb phrase *had pleased*; data from Gibson and Warren (2004)<sup>4</sup>

Sentence	Verb phrase	Reading time	Tenure
'consultant claimed' (80)	had pleased	489	13+4 (17)
'consultant's claim' (81)	had pleased	564	29+18 (47)

tenure. With the start of the phrase *the new proposal* the lines are quite parallel to each other in Figure 5.4.

As mentioned at the beginning of this section, both constructions are matched for the number of intervening discourse referents. Therefore, a strict implementation of DLT would predict no processing differences between (80) and (81) at the VP *had pleased*. However, Gibson and Warren (2004) make the additional assumption of using the intermediate trace in (80) to calculate the DLT distance between the filler and the gap relative to this trace. The intermediate trace shortens the distance in (80), but since there is no such intermediate trace in (81) DLT would predict an increased processing cost for this construction. Alexopoulou and Keller (2007) criticize this additional assumption of using the intermediate trace for the calculation of the DLT values. We follow Alexopoulou and Keller (2007) and assume that DLT predicts no difference between the two constructions.

The mean reading times and the predicted tenure values for the verb phrase *had pleased* are shown in Table 5.14. Tenure predicts a higher complexity for *had pleased* in (81) in which there is no intermediate trace compared to (80), which matches the data obtained by Gibson and Warren (2004).

<sup>4</sup>Gibson and Warren (2004) combined the words *had* and *pleased* into one region of interest and give only the reading times for the whole region. However, tenure makes predictions for each word in the sentence, therefore the two values in the table represent the value for *had* and *pleased* written with a plus sign.

## 6 General Discussion

### 6.1 Conclusions

This dissertation addressed the question of how linguistic structures can be represented in human memory. It proposed a memory-based computational model consisting of an incremental minimalist grammar parser which calculates offline and on-line complexity profiles for various phenomena from the sentence comprehension literature. The presented architecture links grammatical representations stored in memory directly to cognitive behavior by deriving predictions about sentence processing difficulty. The model's assumptions about memory limitations were evaluated using results from six different sentence comprehension experiments.

The comparison between the two approaches showed that both, DLT and tenure, make similar quantitative predictions about effects of processing difficulty for the first five phenomena, but not the sixth. In contrast to the proposed model, DLT is only a theoretical model and not an implemented computational algorithm. Further, DLT is based on the assumption that linear-based distance between head and dependent in terms of discourse referents accounts for processing difficulty. Contrastingly, tenure introduces structural-based distance, in terms of grammar operations by a parser, between the prediction and retrieval of the word in the sentence as the reason for sentence processing difficulty. The last phenomenon presented in Section 5.2.3 directly addressed these different predictions of the complexity metrics and showed that tenure, which is sensitive to subtle differences in the syntactic structure, captured the patterns in the data better than DLT.

We conclude that the syntactic analysis plays a significant role in memory requirements of parsing. An incremental top-down parser based on a grammar formalism easily computes offline and online complexity profiles, which can be used to derive predictions about sentence processing difficulty.

## **6.2 Future Work**

The work presented in this thesis has shown that the computational model combining a parsing algorithm based on a grammar with a complexity metric can account for certain empirical phenomena. Nevertheless, much more areas of research can be explored.

Using a minimalist grammar in a parsing algorithm offers a lot of opportunities for the extension of the functionalities of the algorithm. This thesis focused on syntactic parsing and the syntactic features of minimalist grammars, but minimalist grammars also define semantic features. It would be interesting to see how the extension of the presented algorithm with semantic features increases the explanatory power of the derived complexity metric.

The proposed complexity metrics, average tenure and tenure, account for memory requirements of the parser. This aspect is an integral part of every parser. Hence, it would be interesting to test if the proposed complexity metric makes similar predictions when a different grammar formalism with comparable explanatory power is used.

The version of the algorithm presented here does not consider any ambiguous sentences. It would be interesting to see how the parser could be expanded to a non-deterministic version, using probabilistic weights learned from a corpus to guide its decisions while parsing ambiguous sentences. Additionally, the question that remains to be investigated is: how does the parser handle structures that are ungrammatical or structures that induce a garden-path effect? An example for an ungrammatical structure that is easier to parse than its grammatical counterpart are gram-

mathematical illusions, such as the missing VP effect. How would a parser simulate human behavior for these constructions?

Another direction for future research is to use different empirical data of online tasks than the one used in this thesis. Here, we only used self-paced reading data which basically reflects several sentence comprehension processes in one reading time measure. Due to the limitations of the method, participants are not able to regress to a previous word and they have to coordinate reading with manual button pressing. Taken together these properties could draw on more working memory resources than natural reading. Hence, using data from eye-tracking experiments in the presented architecture could lead to more explicit assumptions about the underlying sentence comprehension processes.

Further, it is necessary to evaluate the predictions of the complexity metric for other languages than English. Work by Kobele et al. (2012) and Kobele et al. (2013) demonstrated that the proposed top-down parser, with a slightly different offline complexity metric as presented here, can account for embedded and cross-serial verb clusters in German and Dutch. It remains to be seen how the predictions of the complexity metric presented in this dissertation can predict processing difficulty in other languages than English. Of particular interest are constructions that can distinguish between linear and structural distance.



## 7 The Grammar Fragment

% file: grammar.pl  
% author: Sabrina Gerth  
% created: 2nd of March 2012

[ ]::[='T', 'C'].	% regular empty complementizer
[ ]::[='T', +wh, 'Rel'].	% wh-hoisting compl.
[that]::[='T', 'Ce'].	% embedding compl.
[the]::[='Rel', 'D', -case].	% relative determiner
[the]::[='N', 'D', -case].	% ordinary determiner
[a]::[='N', 'D', -case].	% ordinary determiner
[five]::[='N', 'D', -case].	% numeral
[who]::[='N', +f, 'D', -case, -wh].	% 'promoting' wh-word

% common nouns

[employee>::['N'].  
[executive>::['N'].  
[manager>::['N'].  
[woman>::['N'].  
[administrator>::['N'].  
[nurse>::['N'].  
[nurse>::[=p\_from, 'N'].  
[medic>::['N'].  
[reporter>::['N'].  
[photographer>::['N'].  
[story>::['N'].  
[consultant>::['N'].  
[proposal>::['N'].  
[claim>::['N'].  
[parishioners>::['N', -f].  
[preacher>::['N'].  
[officesupplies>::['D', -case].  
[it>::['D', -case].

[employee>::['N', -f].  
[executive>::['N', -f].  
[manager>::['N', -f].  
[woman>::['N', -f].  
[administrator>::['N', -f].  
[nurse>::['N', -f].  
[clinic>::['N'].  
[intern>::['N'].  
[reporter>::['N', -f].  
[editor>::['N'].  
[report>::['N'].  
  
[workers>::['N'].  
[claim>::[=p\_about, 'N'].  
[actress>::['N', -f].  
[newsman>::['N'].

% ....with complements

[fact>::[='Ce', 'N'].  
% possessive

['s']::[='N', ='N', 'N'].  
% possessive



---

% preposition is a case assigner (Haegeman p.193)

[from]::[='D', +case, 'Pfrom'].      [for]::[='D', +case, 'Pfor'].

[about]::[='D', +case, 'Pabout'].      [to]::[='D', +case, 'Pto'].

[with]::[='D', +case, 'Pwith'].      [by]::[='D', +case, 'Pby'].

% little p

[ ]::[=>'Pfrom', p\_from].              [ ]::[=>'Pfor', p\_for].

[ ]::[=>'Pabout', p\_about].            [ ]::[=>'Pto', p\_to].

[ ]::[=>'Pwith', p\_with].              [ ]::[=>'Pby', p\_by].

% be ... from

[be]::[=p\_from, 'V'].

% hope for

[hope]::[=p\_for, 'V'].

% sleep with

[sleep]::[=p\_with, 'V'].

% overlooked by

[overlook]::[=p\_by, 'V'].

% send ... to (ditransitive)

[send]::[=p\_to, ='D', +case, 'V'].

% transitive verbs

[hire]::[='D', +case, 'V'].              [worry]::[='D', +case, 'V'].

[see]::[='D', +case, 'V'].              [steal]::[='D', +case, 'V'].

[supervise]::[='D', +case, 'V'].      [scold]::[='D', +case, 'V'].

[hire]::[=tmp, ='D', +case, 'V'].      [please]::[='D', +case, 'V'].

[scandalize]::[='D', +case, 'V'].      [overlook]::[='D', +case, 'V'].

% CP-taking verb

% claim that ...

[claim]::[='Ce', 'V'].

% little v gets the subject

[ ]::[=>'V', ='D', v].

['-ed']::[=>'V', ='D', ven].

['-ing']::[=>'V', ='D', ving].

% tense

['-ed']::[=>v, +case, 'T'].      [ ]::[=>v, +case, 'T'].

['-ed']::[=>'Have', +case, 'T'].      [ ]::[=>'Have', +case, 'T'].

['-ed']::[=>'Be', +case, 'T'].      [ ]::[=>'Be', +case, 'T'].

% auxiliary verbs

[will]::[=v, +case, 'T'].      [have]::[=ven, 'Have'].

[be]::[=ving, 'Be'].      [be]::[=ven, 'Be'].

% adjectives

[new]::['A'].

% adjectives can also left-adjoin onto nouns

['A']»['N'].

% temporal modifiers

[tomorrow]::[tmp].

startCategory('C').

## **8 Result Tables for the Offline Data**

Table 8.1: Word-by-word predictions for nesting constructions RC/SC vs. SC/RC

Measure	Input	The	executive	who	the	fact	that	the
(61)								
DLT		0	1	0	0	1	0	0
Tenure		0	0	0	0	0	0	0
	employee	stole	office	supplies	worried	hired	the	manager
DLT		1	1	1	9	6	0	1
Tenure		0	20	0	22	8	0	0
(62)								
DLT		0	1	0	0	1	0	0
Tenure		0	0	0	0	0	0	0
	manager	hired	stole	office	supplies	worried	the	executive
DLT		1	3	3	1	6	0	1
Tenure		0	8	16	0	25	0	0

**Table 8.2:** Word-by-word predictions for center-embedded vs. right-branching structures

<b>Measure</b>	<b>Input</b>	The	executive	who	the	manager	worried	hired	the	employee
(63)	The	0	1	0	0	1	3	3	0	1
DLT		0	0	0	0	0	8	16	0	0
Tenure										
(64)	The	0	1	1	0	1	0	3	0	1
DLT		0	0	5	0	0	0	6	0	0
Tenure										
(65)	The	0	1	0	0	1	0	0	0	1
DLT		0	0	0	0	0	0	0	0	0
Tenure										
	saw	3	7	5	0	1	0	0	0	0
DLT		19	19	27	0	0	0	0	0	0
Tenure										
(66)	The	0	1	1	0	1	0	3	0	1
DLT		0	0	5	0	0	0	6	0	0
Tenure										
	who	0	3	0	1	0	0	0	0	0
DLT		0	6	0	0	0	0	0	0	0
Tenure										









## Bibliography

- S. Abney and M. Johnson. Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20(3):233–250, 1991.
- A. V. Aho and J. D. Ullman. *The Theory of Parsing, Translation and Compiling*, Vol. I: Parsing. Prentice Hall, Englewood Cliffs, 1972.
- T. Alexopoulou and F. Keller. Locality, cyclicity and resumption: At the interface between grammar and the human parser. *Language*, 83(1):110–160, 2007.
- J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004.
- M. Babyonyshev and E. Gibson. The complexity of nested structures in Japanese. *Language*, 75(3):423–450, 1999.
- W. Badecker and K. Straub. The processing role of structural constraints on the interpretation of pronouns and anaphors. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 28(4):748–769, 2002.
- M. Bader and I. Lasser. German verb-final clauses and sentence processing: Evidence for immediate attachment. In C. Clifton, L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*, pages 225–242. Lawrence Erlbaum, Mahwah, NJ, 1994.
- B. Bartek, R. L. Lewis, S. Vasishth, and M. R. Smith. In search of on-line locality effects in sentence comprehension. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 37(5):1178–1198, 2011.

- T. G. Bever. Cognition and the development of language. *The Cognitive Basis for Linguistic Structures*, 279, 1970.
- M. S. Blaugers and M. D. S. Braine. Short-term memory limitations on decoding self-embedded sentences. *Journal of Experimental Psychology*, 102(4):745–748, 1974.
- M. F. Boston, J. T. Hale, U. Patil, R. Kliegl, and S. Vasishth. Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus. *Journal of Eye Movement Research*, 2(1):1–12, 2008.
- M. F. Boston, J. T. Hale, S. Vasishth, and R. Kliegl. Parallel processing and sentence comprehension difficulty. *Language and Cognitive Processes*, 26(3):301–349, 2011.
- J. Bresnan. *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, 1982.
- N. Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.
- N. Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA, 1965.
- N. Chomsky. Conditions on transformations. In S. Anderson and P. Kiparsky, editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart and Winston, New York, 1973.
- N. Chomsky. *Lectures on Government and Binding*. Foris, Dodrecht, 1981.
- N. Chomsky. *The Minimalist Program*. Massachusetts Institute of Technology, Boston, MA, 1995.
- K. Clackson, C. Felser, and H. Clahsen. Children’s processing of reflexives and pronouns in english: Evidence from eye-movements during listening. *Journal of Memory and Language*, 65: 128–144, 2011.

- C. Clifton and L. Frazier. Comprehending sentences with long-distance dependencies. In G. Carlson and M. Tanenhaus, editors, *Linguistic Structure in Language Processing*, pages 273–317. Kluwer, Dodrecht, The Netherlands, 1989.
- C. Clifton, S. Speer, and S. P. Abney. Parsing arguments: Phrase structure and argument structure as determinants of initial parsing decisions. *Journal of Memory and Language*, 30:251–271, 1991.
- N. Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24:87–185, 2000.
- N. Cowan. *Working Memory Capacity*. Psychology Press, New York, 2005.
- E. A. Cowper. *Constraints on Sentence Complexity: A Model for Syntactic Processing*. PhD thesis, Brown University, Providence, RI, 1976.
- S. Crain and M. Steedman. On not being led up the garden path: The use of context by the psychological syntax processor. In L. K. D. Dowty and A. Zwicky, editors, *Natural Language Parsing: Psycholinguistics, Computational and Theoretical Perspectives*, pages 320–358. Cambridge University Press, Cambridge, 1985.
- M. Crocker and T. Brants. Wide-coverage probabilistic sentence processing. *Journal of Psycholinguistic Research*, 29(6):647–669, 2000.
- F. Cuetos and D. C. Mitchell. Cross-linguistic differences in parsing: Restrictions on the use of the late closure strategy in spanish. *Cognition*, 30:73–105, 1988.
- I. Cunnings and C. Felser. The role of working memory in the processing of reflexives. *Language and Cognitive Processes*, 28(1-2):188–219, 2013.

- V. Demberg and F. Keller. A psycholinguistically motivated version of TAG. In *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9 2008)*, 2008a.
- V. Demberg and F. Keller. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210, 2008b.
- V. Demberg and F. Keller. A computational model of prediction in human parsing: Unifying locality and surprisal effects. In *Proceedings of CogSci*, Amsterdam, 2009.
- V. Demberg, F. Keller, and A. Koller. Incremental, predictive parsing with psycholinguistically motivated tree-adjoining grammar. *Computational Linguistics*, 39(4):1025–1066, 2013.
- B. Dillon, W. Y. Chow, M. Wagers, T. Guo, F. Liu, and C. Phillips. The structure-sensitivity of memory access: evidence from mandarin chinese. *Frontiers in Psychology: Language Sciences*, 5(1025), 2014.
- H. Drenhaus, S. Frisch, and D. Saddy. Processing negative polarity items: When negation comes through the backdoor. In S. Kepser and M. Reis, editors, *Linguistic Evidence – Empirical, Theoretical, and Computational Perspectives.*, pages 145–165. Mouton de Gruyter (Studies in Generative Grammar 85), Berlin, 2005.
- F. Ferreira. The misinterpretation of noncanonical sentences. *Cognitive Psychology*, 47:164–203, 2003.
- F. Ferreira and C. Clifton. The independence of syntactic processing. *Journal of Memory and Language*, 25:348–368, 1986.
- F. Ferreira and J. M. Henderson. Recovery from misanalyses of garden-path sentences. *Journal of Memory and Language*, 30:725–745, 1991.

- F. Ferreira, K. Christianson, and A. Hollingworth. Misinterpretations of garden-path sentences: Implications for models of sentence processing and reanalysis. *Journal of Psycholinguistic Research*, 30(1):3–20, 2001.
- F. Ferreira, K. G. Bailey, and V. Ferraro. Good-enough representations in language comprehension. *Current Directions in Psychological Science*, 11(1):11–15, 2002.
- C. J. Fiebach, M. Schlesewsky, and A. D. Friederici. Separating syntactic memory costs and syntactic integration costs during parsing: The processing of german WH-questions. *Journal of Memory and Language*, 47:250–272, 2002.
- J. D. Fodor. Empty categories in sentence processing. *Language and Cognitive Processes*, 4:155–209, 1989.
- J. D. Fodor and M. F. Garrett. Some reflections on competence and performance. In J. Lyons and R. J. Wales, editors, *Psycholinguistics Papers*. University of Edinburgh Press, Edinburgh, 1966.
- J. D. Fodor and M. F. Garrett. Some syntactic determinants of sentential complexity. *Perception and Psychophysics*, 2:289–296, 1967.
- L. Frazier. *On Comprehending Sentences: Syntactic Parsing Strategies*. PhD thesis, University of Massachusetts, Amherst, MA, 1979.
- L. Frazier. Syntactic complexity. In L. K. D. Dowty and A. Zwicky, editors, *Natural Language Parsing*, pages 129–189. Cambridge University Press, Cambridge, 1985.
- L. Frazier. Syntactic processing: Evidence from dutch. *Natural Language and Linguistic Theory*, 5:515–559, 1987.
- L. Frazier and J. D. Fodor. The sausage machine: A new two-stage parsing model. *Cognition*, 6:291–325, 1978.
- L. Frazier and K. Rayner. Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14:178–210, 1982.

- L. Frazier and K. Rayner. Resolution of syntactic category ambiguities: Eye movements in parsing lexically ambiguous sentences. *Journal of Memory and Language*, 26:505–526, 1987.
- E. Gibson. *A Computational Theory of Human Linguistic Processing: Memory Limitations and Processing Breakdown*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1991.
- E. Gibson. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68:1–76, 1998.
- E. Gibson. The dependency locality theory: A distance-based theory of linguistic complexity. In A. Marantz, Y. Miyashita, and W. O’Neil, editors, *Image, Language, Brain*, pages 95–126. MIT Press, Cambridge, 2000.
- E. Gibson and E. Fedorenko. The need for quantitative methods in syntax and semantics research. *Language and Cognitive Processes*, 28(1-2):88–124, 2013.
- E. Gibson and J. Thomas. The processing complexity of english center-embedded and self-embedded structures. In C. T. Schütze, editor, *Proceedings of the NELS 26 Sentence Processing Workshop*, pages 45–71, Cambridge, MA, 1996. MIT Occasional Papers in Linguistics 9.
- E. Gibson and J. Thomas. The complexity of nested structures in english: Evidence for the syntactic prediction locality theory of linguistic complexity. unpublished Manuscript, 1997.
- E. Gibson and J. Thomas. Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, 14(3):225–248, 1999.
- E. Gibson and T. Warren. Reading-time evidence for intermediate linguistic structure in long-distance dependencies. *Syntax*, 7(1): 55–78, 2004.

- E. Gibson and H.-H. I. Wu. Processing chinese relative clauses in context. *Language and Cognitive Processes*, 28(1-2):125–155, 2013.
- P. C. Gordon, R. Hendrick, and M. Johnson. Memory interference during language processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(6):1411–1423, 2001.
- T. Graf and B. Marcinek. Evaluating evaluation metrics for minimalist parsing. In *Proceedings of the 2014 Workshop on Cognitive Modeling and Computational Linguistics*, pages 28–36, 2014.
- D. Grodner and E. Gibson. Consequences of the serial nature of linguistic input for sentential complexity. *Cognitive Science*, 29: 261–290, 2005.
- L. Haegemann. *Introduction to Government and Binding Theory*. Basil Blackwell, Oxford, second edition, 1994.
- J. T. Hale. A probabilistic early parser as a psycholinguistic model. In *Proceedings of NAACL*, pages 159–166. Association for Computational Linguistics, Pittsburgh, PA, 2001.
- J. T. Hale. The information conveyed by words in sentences. *Journal of Psycholinguistic Research*, 32(2):101–123, 2003a.
- J. T. Hale. *Grammar, Uncertainty and Sentence Processing*. PhD thesis, Baltimore, Maryland, 2003b.
- J. T. Hale. Uncertainty about the rest of the sentence. *Cognitive Science*, 30(4):609–642, 2006.
- H. W. Hamilton and J. Deese. Comprehensibility and subject-verb relations in complex sentences. *Journal of Verbal Learning and Verbal Behavior*, 10:163–170, 1971.
- H. Harkema. *Parsing Minimalist Languages*. Phd thesis, University of California, Los Angeles, 2001.
- J. A. Hawkins. *A Performance Theory of Order and Constituency*. Cambridge University Press, Cambridge, 1994.

- J. A. Hawkins. Processing complexity and filler-gap dependencies across grammars. *Language*, 75(2):244–285, 1999.
- A. Hestvik, E. Bradley, and C. Bradley. Working memory effects of gap-predictions in normal adults: An event-related potentials study. *Journal of Psycholinguistic Research*, 41(6):425–438, 2012.
- F. Hsiao and E. Gibson. Processing relative clauses in chinese. *Cognition*, 90:3–27, 2003.
- P. N. Johnson-Laird. *Mental Models*. Harvard University Press, 1983.
- A. K. Joshi. Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes*, 5(1):1–27, 1990.
- A. K. Joshi, L. S. Levy, and M. Takahashi. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10:136–163, 1975.
- D. Jurafsky. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, 20(2):137–194, 1996.
- M. A. Just and P. A. Carpenter. A theory of reading: From eye fixations to comprehension. *Psychological Review*, 87(4):329–354, 1980.
- M. A. Just and P. A. Carpenter. A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, 99(1):122–149, 1992.
- E. Kaan, A. Harris, E. Gibson, and P. J. Holcomb. The p600 as an index of syntactic integration difficulty. *Language and Cognitive Processes*, 15(2):159–201, 2000.
- R. S. Kayne. *The Antisymmetry of Syntax*. MIT Press, Cambridge, MA, 1994.
- J. Kimball. Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47, 1973.



- J. King and M. A. Just. Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language*, 30:580–602, 1991.
- J. King and M. Kutas. Who did what and when? using word- and clause-level ERPs to monitor working memory usage in reading. *Journal of Cognitive Neuroscience*, 7(3):376–395, 1995.
- G. M. Kobele. *Generating Copies: An Investigation into Structural Identity in Language and Grammar*. PhD thesis, University of California, Los Angeles, 2006.
- G. M. Kobele, C. Retoré, and S. Salvati. An automata-theoretic approach to minimalism. In J. Rogers and S. Kepsner, editors, *Model Theoretic Syntax at 10*, pages 71–80, Dublin, 2007. ESSLLI'07.
- G. M. Kobele, E. Lagrou, F. Engelmann, T. von der Malsburg, R. Musa, S. Gerth, R. van de Vijver, and J. T. Hale. Incremental processing difficulty in cross-serial and nested verb clusters. Poster presented at the 18th AMLaP Conference, Riva del Garda, Italy, 2012.
- G. M. Kobele, S. Gerth, and J. T. Hale. Memory resource allocation in top-down minimalist parsing. In G. Morrill and M.-J. Nederhof, editors, *FG 2012/2013*, volume 8036 of *Lecture Notes in Computer Science*, pages 32–51. Springer, Berlin, 2013.
- L. Konieczny. Locality and parsing complexity. *Journal of Psycholinguistic Research*, 29(6):627–645, 2000.
- N. Kwon. A parsing paradox in head final languages: Head-driven vs. incremental parsing. In E. Efnér and M. Walkow, editors, *Proceedings of the 37th Annual Meeting of the North East Linguistic Society (NELS 37)*. Amherst, MA, University of Massachusetts, 2008.

- N. Kwon, P. C. Gordon, R. Kluender, and M. Polinsky. Cognitive and linguistic factors affecting subject/object asymmetry: An eye-tracking study of prenominal relative clauses in Korean. *Language*, 86(3):546–582, 2010.
- W. J. M. Levelt. A scaling approach to the study of syntactic relations. In G. B. F. D’Arcais and W. Levelt, editors, *Advances in Psycholinguistics*, pages 109–121, New York, 1970. American Elsevier.
- R. Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008.
- R. Levy and F. Keller. Expectation and locality effects in German verb-final structures. *Journal of Memory and Language*, 68(2):199–222, 2013.
- R. Levy, E. Fedorenko, and E. Gibson. The syntactic complexity of Russian relative clauses. *Journal of Memory and Language*, 69(4):461–495, 2013.
- R. L. Lewis. *An Architecturally-based Theory of Human Sentence Comprehension*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1993.
- R. L. Lewis. Interference in short-term memory: The magical number two (or three) in sentence processing. *Journal of Psycholinguistic Research*, 25(1):93–115, 1996.
- R. L. Lewis and S. Vasishth. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29:1–45, 2005.
- R. L. Lewis, S. Vasishth, and J. A. V. Dyke. Computational principles of working memory in sentence comprehension. *Trends in Cognitive Sciences*, 10(10):447–454, 2006.
- S. Lewis and C. Phillips. Aligning grammatical theories and language processing models. *Journal of Psycholinguistic Research: Special Issue on Grammars and Parsers*, under review.

- M. C. MacDonald and M. H. Christiansen. Reassessing working memory: Comment on just and carpenter (1992) and waters and caplan (1996). *Psychological Review*, 109(1):35–54, 2002.
- M. C. MacDonald, N. J. Pearlmutter, and M. S. Seidenberg. Syntactic ambiguity resolution as lexical ambiguity resolution. In C. Clifton, L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*. Erlbaum, Hillsdale, NJ, 1994.
- T. Mainguy. A probabilistic top-down parser for minimalist grammars. ArXiv cs.CL 1010.1826, 2010.
- M. P. Marcus. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge, 1980.
- D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco, 1982.
- W. Marslen-Wilson. Linguistic structure and speech shadowing at very short latencies. *Nature*, 244:522–524, 1973.
- B. McElree. Accessing recent events. In B. H. Ross, editor, *The Psychology of Learning and Motivation*, volume 46. Academic Press, San Diego, 2006.
- B. McElree, S. Foraker, and L. Dyer. Memory structures that subserve sentence comprehension. *Journal of Memory and Language*, 48:67–91, 2003.
- G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 101(2):343–352, 1956.
- G. A. Miller and N. Chomsky. Finitary models of language users. In D. R. Luce, R. R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology*, volume II., pages 419–491. John Wiley, New York, 1963.

- D. C. Mitchell. Sentence parsing. In M. Gernsbacher, editor, *Handbook of Psycholinguistics*. New York: Academic Press, New York, 1994.
- D. C. Mitchell and V. M. Holmes. The role of specific information about the verb in parsing sentences with local structural ambiguity. *Journal of Memory and Language*, 24:542–559, 1985.
- D. C. Mitchell, F. Cuetos, M. M. B. Corley, and M. Brysbaert. Exposure-based models of human parsing: Evidence for the use of coarse-grained (nonlexical) statistical records. *Journal of Psycholinguistic Research*, 24(6):469–488, 1995.
- E. T. Miyamoto and M. Nakamura. Subject/object asymmetries in the processing of relative clauses in japanese. In G. Garding and M. Tsujimura, editors, *Proceedings of the 22nd West Coast Conference on Formal Linguistics*, pages 342–355. Cascadilla Press, Somerville, MA, 2003.
- M. Nakamura and E. T. Miyamoto. Crossed dependencies and plausibility factors in the interpretation of double-gap relative clauses in japanese. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, 13:369–391, 2006.
- J. Nicol and D. A. Swinney. The role of structure in coreference assignment during sentence comprehension. *Journal of Psycholinguistic Research*, 18(1):5–19, 1989.
- J. Nivre. Incrementality in deterministic dependency parsing. In *Incremental Parsing: Bringing Engineering and Cognition Together*, Barcelona, 25th July 2004. Workshop at ACL-2004, Barcelona, Spain.
- W. O’Grady. *Syntactic Development*. University of Chicago Press, Chicago, 1997.
- C. Phillips. Linear order and constituency. *Linguistic Inquiry*, 34(1): 37–90, 2003.

- M. J. Pickering, C. Clifton, and M. W. Crocker. Introduction. In M. W. Crocker, M. J. Pickering, and C. Clifton, editors, *Architectures and Mechanisms for Language Processing*, Cambridge, UK, 1999. Cambridge University Press.
- B. L. Pritchett. Head position and parsing ambiguity. *Journal of Psycholinguistic Research*, 20(3):251–270, 1991.
- O. Rambow and A. K. Joshi. A processing model for free word order languages. In C. Clifton, L. Frazier, and K. Rayner, editors, *Perspectives on sentence processing*, pages 267–301. Lawrence Erlbaum Associates, 1994.
- P. Resnik. Left-corner parsing and psychological plausibility. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING 92)*, Nantes, France, 1993.
- T. J. Ricker, A. M. AuBuchon, and N. Cowan. Working memory. *Wiley Interdisciplinary Reviews in Cognitive Science*, 1:573–585, 2010.
- B. Roark. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276, 2001.
- B. Roark, A. Bachrach, C. Cardenas, and C. Pallier. Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 324–333, Singapore, 2009. Association for Computational Linguistics.
- H. B. Savin and E. Perchonock. Grammatical structure and the immediate recall of english sentences. *Journal of Verbal Learning and Verbal Behavior*, 4:348–353, 1965.
- I. M. Schlesinger. *Sentence Structure and the Reading Process*. Mouton, The Hague, Paris, 1968.

- I. A. Sekerina. Scrambling and processing: Dependencies, complexity, and constraints. In S. Karimi, editor, *Word Order and Scrambling*. Blackwell, Malden, MA, 2003.
- T. J. Slattery, P. Sturt, K. Christianson, M. Yoshida, and F. Ferreira. Lingering misinterpretations of garden path sentences arise from competing syntactic representations. *Journal of Memory and Language*, 69:104–120, 2013.
- M. F. St. John and J. L. McClelland. Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence*, 46:217–257, 1990.
- E. P. Stabler. The finite connectivity of linguistic structure. In C. Clifton, L. Frazier, and K. Rayner, editors, *Perspectives on sentence processing*, pages 303–336. Lawrence Erlbaum, 1994.
- E. P. Stabler. Derivational minimalism. *Lecture Notes in Computer Science*, 1328:68–95, 1997.
- E. P. Stabler. Recognizing head movement. In *Logical Aspects of Computation Linguistics: 4th International Conference*, pages 245–260, NY, 2001. Springer.
- E. P. Stabler. Computational perspectives on minimalism. In C. Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 616–641. Oxford University Press, Oxford, 2011a.
- E. P. Stabler. Top-down recognizers for MCFGs and MGs. In *Proceedings of the 2011 Workshop on Cognitive Modeling and Computational Linguistics*, 2011b.
- E. P. Stabler. Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science*, 5:611–633, 2013.
- A. Staub. Eye movements and processing difficulty in object relative clauses. *Cognition*, 116(71–86), 2010.
- A. Staub and C. Clifton. Syntactic prediction in language comprehension: Evidence from either ... or. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 32(2):425–436, 2006.

- P. Sturt. The time-course of the application of binding constraints in reference resolution. *Journal of Memory and Language*, 48:542–562, 2003.
- P. Sturt and V. Lombardo. Processing coordinated structures: Incrementality and connectedness. *Cognitive Science*, 29(2):291–305, 2005.
- M. K. Tanenhaus and J. C. Trueswell. Sentence comprehension. In J. L. Miller and P. D. Eimas, editors, *Handbook of Perception and Cognition, Vol. 11: Speech, Language and Communication*. Academic Press, 1995.
- M. K. Tanenhaus, M. Spivey-Knowlton, K. M. Eberhard, and J. C. Sedivy. Integration of visual and linguistic information during spoken language comprehension. *Science*, 268:1623–1634, 1995.
- D. J. Townsend and T. G. Bever. *Sentence Comprehension: The Integration of Habits and Rules*. MIT Press, Cambridge, MA, 2001.
- M. J. Traxler, R. K. Morris, and R. E. Seely. Processing subject and object relative clauses: Evidence from eye movements. *Journal of Memory and Language*, 47:69–90, 2002.
- J. C. Trueswell, M. K. Tanenhaus, and S. M. Garnsey. Semantic influences on parsing: Use of thematic role information in syntactic ambiguity resolution. *Journal of Memory and Language*, 33:285–318, 1994.
- M. Ueno and S. M. Garnsey. An ERP study of the processing of subject and object relative clauses in Japanese. *Language and Cognitive Processes*, 23:646–688, 2008.
- J. A. Van Dyke and R. L. Lewis. Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. *Journal of Memory and Language*, 49:285–316, 2003.

- R. P. G. Van Gompel and M. Pickering. Syntactic parsing. In G. Gaskell, editor, *Oxford Handbook of Psycholinguistics*, pages 289–307. Oxford University Press, Oxford, 2007.
- R. P. G. Van Gompel, M. J. Pickering, J. Pearson, and S. P. Liv-  
ersedge. Evidence against competition during syntactic ambi-  
guity resolution. *Journal of Memory and Language*, 52:284–307,  
2005.
- S. Vasishth and H. Drenhaus. Locality effects in German. *Dialogue  
and Discourse*, 1(2):59–82, 2011.
- S. Vasishth and R. L. Lewis. Argument-head distance and process-  
ing complexity: Explaining both locality and antilocality effects.  
*Language*, 82(4):767–794, 2006.
- S. Vasishth, S. Brüssow, R. L. Lewis, and H. Drenhaus. Processing  
polarity: How the ungrammatical intrudes on the grammatical.  
*Cognitive Science*, 32(4):685–712, 2008.
- S. Vasishth, K. Suckow, R. L. Lewis, and S. Kern. Short-term forget-  
ting in sentence comprehension: Crosslinguistic evidence from  
verb-final structures. *Language and Cognitive Processes*, 25(4):533–  
567, 2010.
- T. Vosse and G. Kempen. Syntactic structure assembly in human  
parsing: A computational model based on competitive inhibi-  
tion and a lexicalist grammar. *Cognition*, 75:105–143, 2000.
- T. Vosse and G. Kempen. The unification space implemented  
as a localist neural net: Predictions and error-tolerance in a  
constraint-based parser. *Cognitive Neurodynamics*, 3:331–346,  
2009.
- M. Wagers. Memory mechanisms for wh-dependency forma-  
tion and their implications for islandhood. In J. Sprouse and  
N. Hornstein, editors, *Experimental Syntax and Island Effects*,  
pages 161–185. Cambridge University Press, 2013.



- M. Wagers, E. F. Lau, and C. Phillips. Agreement attraction in comprehension: Representations and processes. *Journal of Memory and Language*, 61(2):206–237, 2009.
- E. Wanner and M. Maratsos. An ATN approach to comprehension. In M. Halle, J. Bresnan, and G. A. Miller, editors, *Linguistic Theory and Psychological Reality*. MIT Press, Cambridge, MA, 1978.
- T. Warren and E. Gibson. The influence of referential processing on sentence complexity. *Cognition*, 85:79–112, 2002.
- T. Warren and E. Gibson. Effects of NP type in reading cleft sentences in english. *Language and Cognitive Processes*, 20(6):751–767, 2005.
- M. Xiang, B. Dillon, and C. Phillips. Illusory licensing effects across dependency types: ERP evidence. *Brain and Language*, 108:40–55, 2009.
- V. H. Yngve. A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society*, 104:444–466, 1960.
- D. H. Younger. Recognition and parsing of context-free languages in time  $n^{3*}$ . *Information and Control*, 10:189–208, 1967.





This dissertation addresses the question of how linguistic structures can be represented in working memory. We propose a memory-based computational model that derives offline and online complexity profiles in terms of a top-down parser for minimalist grammars (Stabler, 2011). The complexity metric reflects the amount of time an item is stored in memory. The presented architecture links grammatical representations stored in memory directly to the cognitive behavior by deriving predictions about sentence processing difficulty.

Results from five different sentence comprehension experiments were used to evaluate the model's assumptions about memory limitations. The predictions of the complexity metric were compared to the locality (integration and storage) cost metric of Dependency Locality Theory (Gibson, 2000). Both metrics make comparable offline and online predictions for four of the five phenomena. The key difference between the two metrics is that the proposed complexity metric accounts for the structural complexity of intervening material. In contrast, DLT's integration cost metric considers the number of discourse referents, not the syntactic structural complexity.

We conclude that the syntactic analysis plays a significant role in memory requirements of parsing. An incremental top-down parser based on a grammar formalism easily computes offline and online complexity profiles, which can be used to derive predictions about sentence processing difficulty.

ISSN 2190-4545  
ISBN 978-3-86956-321-3

